
IBM eServer p5 570

Model 9117-570

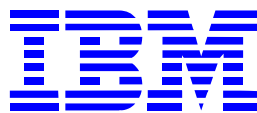
Using

AIX 5L Version 5.3

and

DB2 Universal Database 8.1

TPC BenchmarkTM C
Full Disclosure Report



Second Edition

April 10, 2006

Special Notices

The following terms used in this publication are trademarks of **International Business Machines** Corporation in the United States and/or other countries:

IBM eServer pSeries

IBM eServer xSeries

AIX

IBM

DB2, DB2 UDB, DB2 Universal Database

The following terms used in this publication are trademarks of other companies as follows:

TPC Benchmark, TPC-C, and tpmC are trademarks of the Transaction Processing Performance Council

Microsoft Windows 2000 server and COM+ are registered trademarks of Microsoft Corporation

First Edition: July 12, 2004

Second Edition: April 10, 2006

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

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

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

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

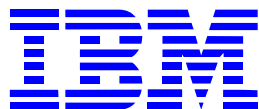
Request for additional copies of this document should be sent to the following address:

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

© Copyright International Business Machines Corporation, 2004. All rights reserved.

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

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



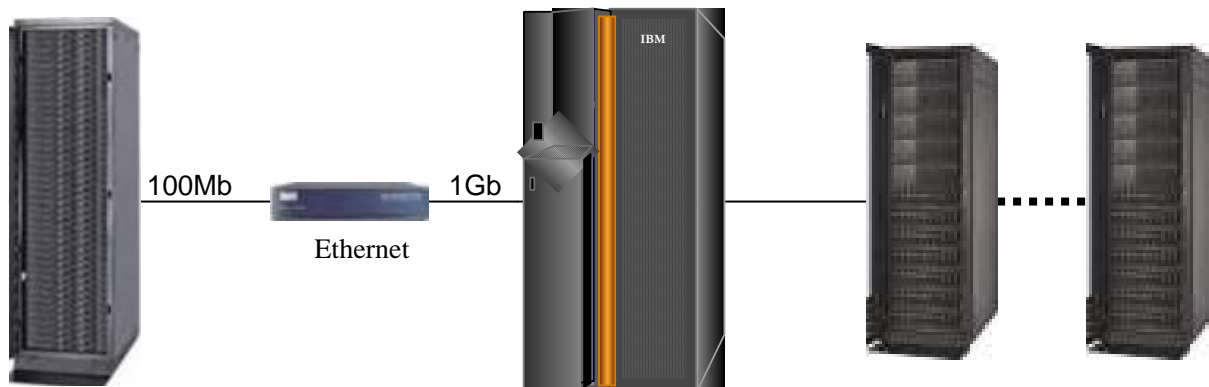
**IBM eServer p5 570
Model 9117-570**

TPC-C Rev. 5.3

Report Date: July 12, 2004

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
\$4,004,491 USD	809,144.09	\$4.95 USD	September 30, 2004	
Processor Chips/Cores/Threads	Database Manager	Operating System	Other Software	No. Users
8/16/32	DB2 UDB 8.1	AIX 5L V5.3	Microsoft COM+	640,000

SUT



32 Clients
 IBM eServer™ x335
 2x3.2GHz Intel® Xeon™
 1MB L3 Cache
 2GB Memory
 1 80GB Internal Drive
 2 Integrated 10/100/1000

IBM® eServer™ p5 570
 8 Processor chips with
 16x 1.9GHz POWER5™ Cores
 36MB L3 Cache per Chip
 512GB Memory
 8 Internal Ultra4 SCSI
 3 36GB Internal SCSI Drives
 2 72GB Internal SCSI Drives
 22 IBM FAStT 2Gb FC Adapters
 8 Integrated Dual port 10/100/1000 Ethernet

Storage
 11 IBM FAStT 900 Storage Servers
 124 IBM FAStT 700 Enclosures
 1600 36.4GB 15K RPM Drives
 40 73.4GB 15K RPM Drives

System Components	Each of the 32 Clients		Server	
	Quantity	Description	Quantity	Description
Processors	2	3.2GHz 1MB L3 Xeon Processor	16	1.9GHz POWER5™
Memory	4	512 MB	16	32GB
Disk Controllers	1	EIDE	8 22 11	Integrated Ultra4SCSI FAStT 2Gb FC IBM FAStT900 Controllers
Disk Drives	1	80 GB	1600 40 3 2	36.4GB 15K RPM FC 73.4GB 15K RPM FC 36.4GB 10K RPM SCSI 73.4GB 10K RPM SCSI
Total Storage		2560 GB		614,320 GB
Terminals	1	System Console	1	System Console



IBM eServer p5 570 Model 9117-570

TPC-C Rev. 5.3

Report Date: July 12, 2004

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
IBM eServer p5 570 Acnhor Card VPD	9117-570	1	1,000	1	1,000	92,010
Headless Enclosure	7879	1	350	4	1,400	
I/O Backplane	7866	1	4,100	4	16,400	
System Midplane	7867	1	500	4	2,000	
SCSI disk backplane	7868	1	1,200	4	4,800	
Power Distribution Backplane	7870	1	200	4	800	
Serial Port Riser Card	7878	1	100	4	400	
Power Supply	7888	1	800	8	6,400	
L4 1.9 Processor 0/2Way	7832	1	10,900	8	87,200	
1-way Activation for 7832	7898	1	21,800	8	174,400	
1-way Activation for 7832 - \$0 Priced Value Pack	8454	1		8		
Media Backplane	7869	1	140	1	140	
CEC Backplane	7865	1	1,200	4	4,800	
L4 CPU Regulator	7875	1	1,100	12	13,200	
L4 FSP Card	7881	1	650	1	650	
IDE DVDROM, Slim Line	2640	1	378	1	378	
32GB (4x8000MB)	4492	1	103,000	16	1,648,000	
DASD 73.4 10K rpm	3274	1	1,400	2	2,800	
DASD 36.4 10K rpm	3273	1	750	3	2,250	
Line Cords	6458	1	14	8	112	
Op Panel	1846	1	150	1	150	
SMP Flex Cable - 4 encl	1849	1	13,500	1	13,500	
FSP Flex Cable - 4 encl	1859	1	5,600	1	5,600	
Hardware Management Console	7315-C03	1	2,535	1	2,535	144
HMC Cable	8121	1	62	1	62	
Quiet Touch Keyboard	8800	1	83	1	83	
Mouse	8841	1	62	1	62	
Model D20 I/O Drawer	7311-D11	1	4,461	4	17,844	11,136
Remote I/O Cable, 1.2M	3146	1	417	4	1,668	
SPCN 2m Cable: Drawer to Drawer	6001	1	25	5	125	
AC Power Supply, 250W	6278	1	375	8	3,000	
RIO-2 Ports to I/O Planar Riser Card	6431	1	900	4	3,600	
Rack Mount enclosure	7311	1	417	2	834	
2 Gigabit Fibre Channel PCI-X Adapter	5716	1	2,267	22	49,874	
Rack Model T00 - 6098, 6107	7014-T00	1	3,370	1	3,370	768
				Subtotal	2,069,437	104,058
Storage						
FAST EXP 700	1740-1RU	1	6,000	124	744,000	
Short Wave SFP	2210	1	499	518	258,482	
2Gb FC, 36.4GB/15K Drive	5212	1	1,115	1,600	1,784,000	
2Gb FC, 73.4GB/15K Disk Module	5213	1	2,099	40	83,960	
FAST900 Storage Server	1742-90U	1	66,500	11	731,500	
Fiber Cable 25m	5625	1	189	22	4,158	
Fiber Cable 1m	5601	1	79	228	18,012	
Fiber Cable 5m	5605	1	129	20	2,580	
2Gb Mini HUB	3507	1	899	22	19,778	
FAST Storage Manager V8.4 upgrade	1742-7109	1	2,999	1	2,999	
3 Year Warranty Service Upgrade 1740-1RU 24x7x4	694225B1813	1	760	124		94,240
3 Year Warranty Service Upgrade 1742-90U 24x7x4	694225B1915	1	1,087	22		23,914
				Subtotal	3,649,469	118,154



**IBM eServer p5 570
Model 9117-570**

TPC-C Rev. 5.3

Report Date: July 12, 2004

Server Software

AIX 5.3 (media only)	5962-A5L	1	50	1	50	
AIX Software	5765-AIX	1	1,225	16	19,600	
AIX Software Maintenance (3Y)	5773-SM3	1	1,958	16		31,328
AIX Software Maintenance 24x7 Upgrade (3Y)	0468	1	496	16		7,936
HMC Software SUB	5639-SB1	1	275	3	33	825
HMC Software Support	5639-ST1	1	525	3		1,575
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	515	1	515	
C for AIX user annual SW maint renewal	E1A1FLL	1	103	2		206
DB2 Enterprise Server Edition Proc Lic/1 yr Maint.	D518GLL	1	19,755	16	316,080	
DB2 Enterprise Server Ed Proc Maint Renew	E00BILL	1	940	32		30,080
				Subtotal	336,278	71,950

Client Hardware and Software

x335 w 3.2GHz/1024KB Xeon DP, one 80GB drive	864762X	1	2,679	32	85,728	14,400
3.2GHz 533 MHz 1MB L3 Cache Xeon Processor	13N0661	1	1,549	32	49,568	
512MB PC2100 CL2.5 ECC DDR SDRAM RDIMM	33L5038	1	219	64	14,016	
Microsoft Windows 2000 Server (Preloaded)	09N9982	1	799	32	25,568	
IBM ServicePac for xSeries	29R5396	1	1,375	32		44,000
NetBay42 Standard Rack	9306-421	1	1,439	12	17,268	2,016
xSeries Cable Chain Technology Cable Kit	06P4792	1	54	1	54	
IBM Sleek 2-Button Mouse - (PS/2)	28L3673	1	15	1	15	
Preferred Pro Full Size PS/2 Keyboard	31P7415	1	29	1	29	
E54 15" monitor/Keyboard/Mouse	W9SP47N	1	119	1	119	
				Subtotal	192,365	60,416

Third Party Hardware/Software

Visual Studio .NET Professional	528577	2	1,016	1	1,016	
Netgear FS750 48 ports w/Gigabit ports	500969	2	486	3	1,458	
				Subtotal	2,474	
				IBM Discounts	-2,492,379	-107,732
				Total	3,757,645	246,846

Three-Year Cost of Ownership 4,004,491

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

tpmC 809,144

Pricing Sources:

\$/tpmC 4.95

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

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

2. CDW.com

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

Numerical Quantities Summary for the IBM eServer p5 570 Model 9117-570

MQTH, computed Maximum Qualified Throughput: 809,144.09 tpmC

<u>Response Times (in seconds)</u>	<u>90th %</u>	<u>Average</u>	<u>Maximum</u>
New Order	0.26	0.19	5.66
Payment	0.26	0.19	5.72
Order-Status	0.27	0.19	5.63
Delivery (interactive)	0.14	0.13	5.14
Delivery (deferred)	0.23	0.15	4.72
Stock-Level	0.33	0.22	5.80
Menu	0.15	0.12	5.18

Response time delay added for emulated components was 0.1 seconds

<u>Transaction Mix, in percent of total transactions</u>	<u>Percent</u>
New Order	44.90%
Payment	43.03%
Order-Status	4.02%
Delivery	4.02%
Stock-Level	4.02%

<u>Keying/Think Times (in seconds)</u>	<u>Min.</u>	<u>Average</u>	<u>Max.</u>
New Order	18.00/0.01	18.01/12.02	19.05/120.25
Payment	3.00/0.01	3.01/12.02	3.91/120.30
Order-Status	2.00/0.01	2.01/10.01	2.76/100.15
Delivery	2.00/0.01	2.01/5.03	2.49/50.23
Stock-Level	2.00/0.01	2.01/5.02	2.93/50.23

Test Duration

Ramp-up Time	44 minutes
Measurement interval	2 hours 30 minutes
Transactions during measurement interval (all types)	270,288,627
Ramp-down time	20 minutes

Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

Table of Content

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

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

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.3 dated April, 2004, for measurements on the IBM eServer p5 570 Model 9117-570. The software used on the IBM eServer p5 570 Model 9117-570 includes AIX 5L Version 5.3 operating system, DB2 UDB 8.1 database manager. Microsoft COM+ is used as transaction manager.

IBM eServer p5 570 Model 9117-570

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM eServer p5 570 Model 9117-570	DB2 UDB 8.1	AIX 5L Version 5.3

Total System Cost	TPC-C Throughput	Price/Performance
<ul style="list-style-type: none">• Hardware• Software• 3 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
\$4,004,491 USD	809,144.09	\$4.95 USD

Preface

TPC Benchmark™ C Standard Specification was developed by the Transaction Processing Performance Council (TPC). It was released on August 13, 1992 and updated with revision 5.3 in April 2004.

This is the full disclosure report for benchmark testing of the IBM eServer p5 570 Model 9117-570 and DB2 UDB 8.1 according to the TPC Benchmark™ C Standard Specification.

TPC Benchmark™ C exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing a mixture of read-only and update intensive transactions. This is a complex OLTP application environment exercising a breadth of system components associated by such environments characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Data bases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention on data access and update

This benchmark defines four on-line transactions and one deferred transaction, intended to emulate functions that are common to many OLTP applications. However, this benchmark does not reflect the entire range of OLTP requirements. The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-C should not be used as a substitute for a specific customer application benchmarks when critical capacity planning and/or product evaluation decisions are contemplated.

The performance metric reported by TPC-C is a “business throughput” measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

0 General Items

0.1. Application Code Disclosure

The application program (as defined in Clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input and output functions.

Appendix A contains the eServer pSeries application code for the five TPC Benchmark™ C transactions. Appendix D contains the terminal functions and layouts.

0.2. Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

This benchmark was sponsored by **International Business Machines Corporation.**

0.3. Parameter Settings

Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:

- *Data Base tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*

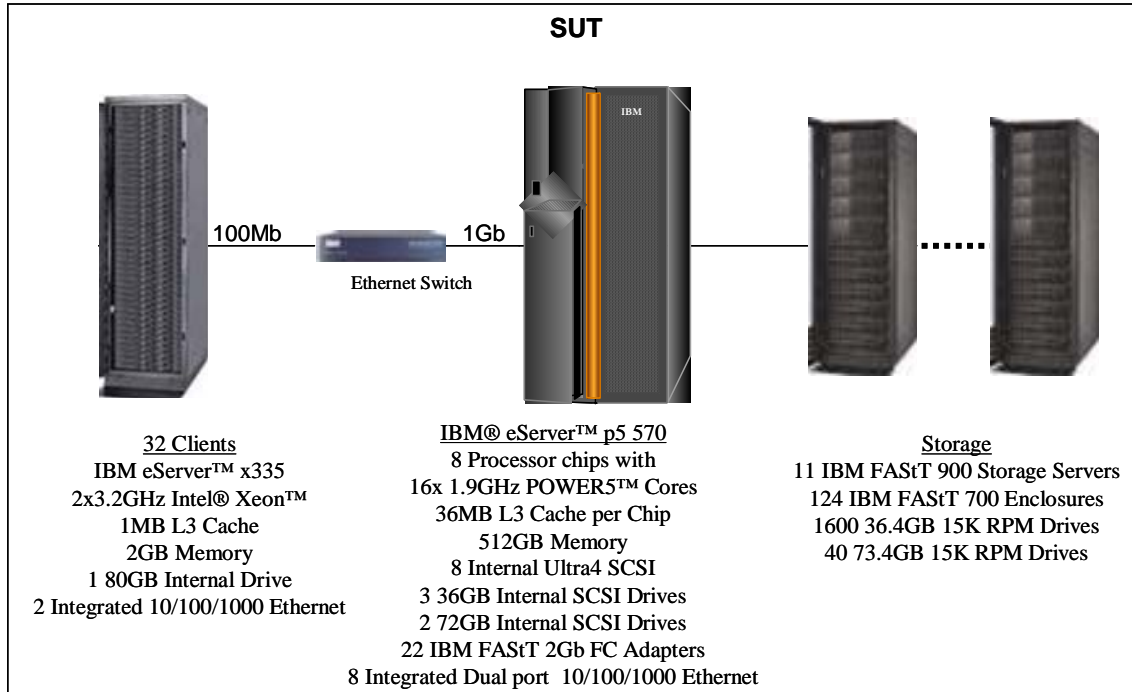
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

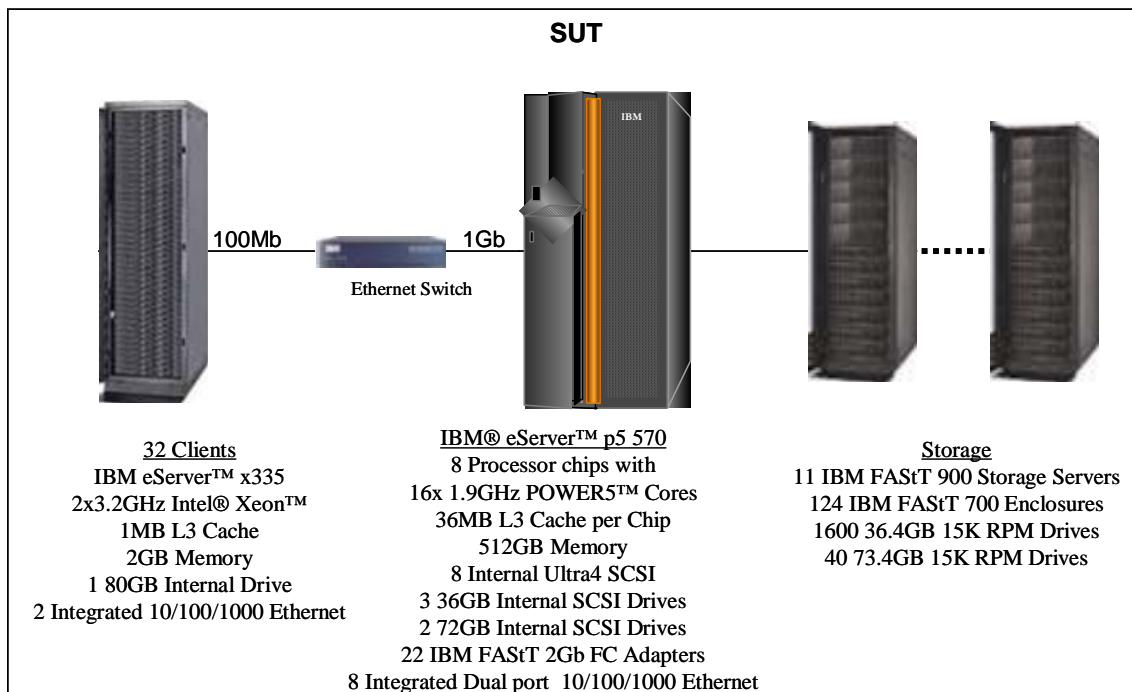
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test*
- *Number and type of disk units (and controllers, if applicable)*
- *Number of channels or bus connections to disk units, including the protocol type*
- *Number of LAN (e.g. Ethernet) connections, including routers, work stations, terminals, etc, that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8)*
- *Type and run-time execution location of software components (e.g. DBMS, client processes, transaction monitors, software drivers, etc)*

IBM eServer p5 570 Model 9117-570 Benchmark Configuration



IBM eServer p5 570 Model 9117-570 Priced Configuration



1 Clause 1: Logical Data Base Design Related Items

1.1. Table Definitions

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

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

1.2. Database Organization

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

Physical space was allocated to DB2 UDB on the server disks according to the details provided in Appendix C.

1.3. Insert and/or Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT data base implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

There were no restrictions on insert and/or delete operations to any of the tables. The space required for an additional five percent of the initial table cardinality was allocated to DB2 UDB and priced as static space.

The insert and delete functions were verified by the auditor. In addition, the auditor verified that the primary key for each database table could be updated outside the range of its initial partition.

1.4. Horizontal or Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark, any such partitioning must be disclosed.

All tables but ITEM were horizontally partitioned into multiple tables.

Each (STOCK, CUSTOMER, ORDERS, ORDERLINE) table partition contains data associated with a range of 1600 warehouses.

Each (WAREHOUSE, DISTRICT, HISTORY, NEWORDER) table partition contains data associated with a range of 8000 warehouses.

A view was created over all partitioned tables to provide full transparency of data manipulation.

No tables were replicated.

2 Clause 2: Transaction & Terminal Profiles Related Items

2.1. Verification for the Random Number Generator

The method of verification for the random number generation must be disclosed.

The `srandom()`, `getpid()` and `gettimeofday()` functions are used to produce unique random seeds for each driver. The drivers use these seeds to seed the `srand()`, `srandom()` and `srand48()` functions. Random numbers are produced using wrappers around the standard system random number generators.

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

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

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

2.2. Input/Output Screens

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

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

2.3. Priced Terminal Features

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

The emulated workstations, IBM eServer x335, are commercially available and support all of the requirements in Clause 2.2.2.4.

2.4. Presentation Managers

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

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

2.5. Home and Remote Order-lines

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

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

2.6. New-Order Rollback Transactions

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

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

2.7. Number of Items per Order

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

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

2.8. Home and Remote Payment Transactions

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

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

2.9. Non-Primary Key Transactions

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

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

2.10. Skipped Delivery Transactions

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

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

2.11. Mix of Transaction Types

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

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

New Order	IBM eServer p5 570 Model 9117-570
Percentage of Home order lines	99.01%
Percentage of Remote order lines	0.99%
Percentage of Rolled Back Transactions	0.99%
Average Number of Items per order	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	60.00%
Percentage of Order-Status using C_LAST	59.99%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.90%
Payment	43.03%
Order-Status	4.02%
Delivery	4.02%
Stock-Level	4.02%

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

2.12. Queuing Mechanism of Delivery

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

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

3 Clause 3: Transaction and System Properties

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

All ACID tests were conducted according to specification.

3.1. Atomicity Requirements

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

3.1.1. Atomicity of Completed Transaction

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

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

1. The balance, BALANCE_1, was retrieved from the CUSTOMER table for a random Customer, District and Warehouse combination.
2. The Payment transaction was executed and committed for the Customer, District, and Warehouse combination used in step 1.
3. The balance, BALANCE_2, was retrieved again for the Customer, District, and Warehouse combination used in step 1 and step 2. It was verified that BALANCE_1 was greater than BALANCE_2 by the amount of the Payment transaction.

3.1.2. Atomicity of Aborted Transactions

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

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

1. The Payment application code was implemented with a Perl script that allowed the transaction to be rolled back rather than committed.
2. The balance, BALANCE_3, was retrieved from the Customer table for the same Customer, District, and Warehouse combination used in the completed Payment transaction Atomicity test.
3. The Payment transaction was executed for the Customer, District and Warehouse used in step 2. Rather than commit the transaction, the transaction was rolled back.
4. The balance, BALANCE_4 was retrieved again for the Customer, District, and Warehouse combination used in step 2. It was verified that BALANCE_4 was equal to BALANCE_3, demonstrating that there were no remaining effects of the rolled back Payment transaction.

3.2. Consistency Requirements

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

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

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

1. The sum of balances (d_ytd) for all Districts within a specific Warehouse is equal to the balance (w_ytd) of that Warehouse.
2. For each District within a Warehouse, the next available Order ID (d_next_o_id) minus one is equal to the most recent Order ID [max(o_id)] for the Order table associated with the preceding District and Warehouse.

Additionally, that same relationship exists for the most recent Order ID [max(o_id)] for the New Order table associated with the same District and Warehouse. Those relationships can be illustrated as follows:

$$d_next_o_id - 1 = \max(o_id) = \max(no_o_id)$$

where (d_w_id = o_w_id = no_w_id) and (d_id = o_d_id = no_d_id)

3. For each District within a Warehouse, the value of the most recent Order ID [max(no_o_id)] minus the first Order ID [min(no_o_id)] plus one, for the New Order table associated with the District and Warehouse equals the number of rows in that New Order table. That relationship can be illustrated as follows:

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

4. For each District within a Warehouse, the sum of Order Line counts [sum(o_ol_cnt)] for the Order table associated with the District equals the number of rows in the Order Line table associated with the same District. That relationship can be illustrated as follows:

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

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.

3.3. Isolation Requirements

Operations of concurrent data base transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

The benchmark specification defines nine tests to demonstrate the property of transaction isolation. The tests, described in Clauses 3.4.2.1 – 3.4.2.9 were all successfully executed using a series of scripts. Case A was observed during the execution of Isolation Tests 7-9.

3.4. Durability Requirements

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure data base consistency after recovery from any one of the failures listed in Clause 3.5.3

3.4.1. Permanent Unrecoverable Failure of any Single Durable Medium

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

Failure of Log Disk and Log Cache:

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

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 10 minutes.
3. One of the disks containing the transaction log was removed. Since the disk was RAID-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.

4 Clause 4: Scaling and Data Base Population Related Items

4.1. Cardinality of Tables

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

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

All tables are based on 64,000 warehouses, the number of active warehouses during the benchmark.

Table Name	Number of Rows
Warehouse	64,000
District	640,000
Customer	1,920,000,000
History	1,920,000,000
Order	1,920,000,000
New Order	576,000,000
Order Line	19,200,069,920
Stock	6,400,000,000
Item	100,000

Table 4-1: Initial Cardinality of Tables

4.2. Distribution of Tables and Logs

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

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

There are 2 storage adapters for the log LV. Each adapter has 2 logical disks (hdisks) and the log LV is striped across the 4 hdisks.

Each hdisk is configured as a RAID10 disk array, with 10 physical disks.

There is a total of 40 physical disks for the logs, and each physical disk has a capacity of 73.4 GB

There are 480 Logical Volumes (LV) for the tables.

There are 10 pairs of storage adapters for the tables. Each adapter has 4 hdisks and each LV is spread across 2 hdisks (1 hdisk on each adapter of a pair).

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

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

There is a total of 1600 data disks and each physical disk has a capacity of 36.4 GB drives.

Each LV correspond to one DB2 container.

Each partition within a partitioned table is made of either one DB2 container (STOCK, CUSTOMER, ORDERS, ORDERLINE)

or 5 DB2 containers (WAREHOUSE, DISTRICT, HISTORY, NEWORDER) so that the corresponding view spans all 20 adapters (ie all 80 hdisks).

The ITEM table, which is not partitioned, is made of 40 DB2 containers and also spans all 20 adapters.

4.3. Data Base Model Implemented

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

The database manager used for this testing was DB2 UDB 8.1. 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.

4.4. Partitions/Replications Mapping

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

The Warehouse, District, Customer, Order, Order-Line, New Order, History and Stock tables were horizontally partitioned into multiple tables. The specifics of the distribution of partitioned and non-partitioned tables across the physical media can be found in Table 4-2.

DATA DISTRIBUTION		
HDISK	ADAPTER	LOGICAL VOLUMES
hdisk62 hdisk63	fcs41	db2loglv
hdisk64 hdisk65	fcs42	
hdisk14	fcs2	D1F01V1ITEM, D1F01V1WAE, D1F01V1DIST, D1F01V1CSTI, D1F01V1NORA, D1F01V1ORL, D1F01V1STK, D1F01V1CST, D1F01V1ORDI, D1F01V1ORD, D1F01V1HIST, D1F01V1NORB
hdisk15	fcs3	
hdisk16	fcs2	D1F01V2ITEM, D1F01V2WARE, D1F01V2DIST, D1F01V2CSTI, D1F01V2NORA, D1F01V2ORL, D1F01V2STK, D1F01V2CST, D1F01V2ORDI, D1F01V2ORD, D1F01V2HIST, D1F01V2NORB
hdisk17	fcs3	
hdisk18	fcs2	D1F01V3ITEM, D1F01V3WARE, D1F01V3DIST, D1F01V3CSTI, D1F01V3NORA, D1F01V3ORL, D1F01V3STK, D1F01V3CST, D1F01V3ORDI, D1F01V3ORD, D1F01V3HIST, D1F01V3NORB
hdisk19	fcs3	
hdisk20	fcs2	D1F01V4ITEM, D1F01V4WARE, D1F01V4DIST, D1F01V4CSTI, D1F01V4NORA, D1F01V4ORL, D1F01V4STK, D1F01V4CST, D1F01V4ORDI, D1F01V4ORD, D1F01V4HIST, D1F01V4NORB
hdisk21	fcs3	
hdisk22	fcs4	D1F02V1ITEM, D1F02V1WARE, D1F02V1DIST, D1F02V1CSTI, D1F02V1NORA, D1F02V1ORL, D1F02V1STK, D1F02V1CST, D1F02V1ORDI, D1F02V1ORD, D1F02V1HIST, D1F02V1NORB
hdisk23	fcs5	
hdisk24	fcs4	D1F02V2ITEM, D1F02V2WARE, D1F02V2DIST, D1F02V2CSTI, D1F02V2NORA, D1F02V2ORL, D1F02V2STK, D1F02V2CST, D1F02V2ORDI, D1F02V2ORD, D1F02V2HIST, D1F02V2NORB
hdisk25	fcs5	
hdisk26	fcs4	D1F02V3ITEM, D1F02V3WARE, D1F02V3DIST, D1F02V3CSTI, D1F02V3NORA, D1F02V3ORL, D1F02V3STK, D1F02V3CST, D1F02V3ORDI, D1F02V3ORD, D1F02V3HIST, D1F02V3NORB
hdisk27	fcs5	
hdisk28	fcs4	D1F02V4ITEM, D1F02V4WARE, D1F02V4DIST, D1F02V4CSTI, D1F02V4NORA, D1F02V4ORL, D1F02V4STK, D1F02V4CST, D1F02V4ORDI, D1F02V4ORD, D1F02V4HIST, D1F02V4NORB
hdisk29	fcs5	
hdisk30	fcs6	D1F03V1ITEM, D1F03V1WARE, D1F03V1DIST, D1F03V1CSTI, D1F03V1NORA, D1F03V1ORL, D1F03V1STK, D1F03V1CST, D1F03V1ORDI, D1F03V1ORD, D1F03V1HIST, D1F03V1NORB
hdisk31	fcs7	
hdisk32	fcs6	D1F03V2ITEM, D1F03V2WARE, D1F03V2DIST, D1F03V2CSTI, D1F03V2NORA, D1F03V2ORL, D1F03V2STK, D1F03V2CST, D1F03V2ORDI, D1F03V2ORD, D1F03V2HIST, D1F03V2NORB
hdisk33	fcs7	
hdisk34	fcs6	D1F03V3ITEM, D1F03V3WARE, D1F03V3DIST, D1F03V3CSTI, D1F03V3NORA, D1F03V3ORL, D1F03V3STK, D1F03V3CST, D1F03V3ORDI, D1F03V3ORD, D1F03V3HIST, D1F03V3NORB
hdisk35	fcs7	

hdisk36	fcs6	D1F03V4ITEM, D1F03V4WARE, D1F03V4DIST, D1F03V4CSTI, D1F03V4NORA, D1F03V4ORL, D1F03V4STK, D1F03V4CST, D1F03V4ORDI, D1F03V4ORD, D1F03V4HIST, D1F03V4NORB
hdisk37	fcs7	
hdisk38	fcs8	D1F04V1ITEM, D1F04V1WARE, D1F04V1DIST, D1F04V1CSTI, D1F04V1NORA, D1F04V1ORL, D1F04V1STK, D1F04V1CST, D1F04V1ORDI, D1F04V1ORD, D1F04V1HIST, D1F04V1NORB
hdisk39	fcs9	
hdisk40	fcs8	D1F04V2ITEM, D1F04V2WARE, D1F04V2DIST, D1F04V2CSTI, D1F04V2NORA, D1F04V2ORL, D1F04V2STK, D1F04V2CST, D1F04V2ORDI, D1F04V2ORD, D1F04V2HIST, D1F04V2NORB
hdisk41	fcs9	
hdisk42	fcs8	D1F04V3ITEM, D1F04V3WARE, D1F04V3DIST, D1F04V3CSTI, D1F04V3NORA, D1F04V3ORL, D1F04V3STK, D1F04V3CST, D1F04V3ORDI, D1F04V3ORD, D1F04V3HIST, D1F04V3NORB
hdisk43	fcs9	
hdisk44	fcs8	D1F04V4ITEM, D1F04V4WARE, D1F04V4DIST, D1F04V4CSTI, D1F04V4NORA, D1F04V4ORL, D1F04V4STK, D1F04V4CST, D1F04V4ORDI, D1F04V4ORD, D1F04V4HIST, D1F04V4NORB
hdisk45	fcs9	
hdisk46	fcs10	D1F05V1ITEM, D1F05V1WARE, D1F05V1DIST, D1F05V1CSTI, D1F05V1NORA, D1F05V1ORL, D1F05V1STK, D1F05V1CST, D1F05V1ORDI, D1F05V1ORD, D1F05V1HIST, D1F05V1NORB
hdisk47	fcs11	
hdisk48	fcs10	D1F05V2ITEM, D1F05V2WARE, D1F05V2DIST, D1F05V2CSTI, D1F05V2NORA, D1F05V2ORL, D1F05V2STK, D1F05V2CST, D1F05V2ORDI, D1F05V2ORD, D1F05V2HIST, D1F05V2NORB
hdisk49	fcs11	
hdisk50	fcs10	D1F05V3ITEM, D1F05V3WARE, D1F05V3DIST, D1F05V3CSTI, D1F05V3NORA, D1F05V3ORL, D1F05V3STK, D1F05V3CST, D1F05V3ORDI, D1F05V3ORD, D1F05V3HIST, D1F05V3NORB
hdisk51	fcs11	
hdisk52	fcs10	D1F05V4ITEM, D1F05V4WARE, D1F05V4DIST, D1F05V4CSTI, D1F05V4NORA, D1F05V4ORL, D1F05V4STK, D1F05V4CST, D1F05V4ORDI, D1F05V4ORD, D1F05V4HIST, D1F05V4NORB
hdisk53	fcs11	
hdisk54	fcs13	D1F06V1ITEM, D1F06V1WARE, D1F06V1DIST, D1F06V1CSTI, D1F06V1NORA, D1F06V1ORL, D1F06V1STK, D1F06V1CST, D1F06V1ORDI, D1F06V1ORD, D1F06V1HIST, D1F06V1NORB
hdisk55	fcs12	
hdisk56	fcs13	D1F06V2ITEM, D1F06V2WARE, D1F06V2DIST, D1F06V2CSTI, D1F06V2NORA, D1F06V2ORL, D1F06V2STK, D1F06V2CST, D1F06V2ORDI, D1F06V2ORD, D1F06V2HIST, D1F06V2NORB
hdisk57	fcs12	
hdisk58	fcs13	D1F06V3ITEM, D1F06V3WARE, D1F06V3DIST, D1F06V3CSTI, D1F06V3NORA, D1F06V3ORL, D1F06V3STK, D1F06V3CST, D1F06V3ORDI, D1F06V3ORD, D1F06V3HIST, D1F06V3NORB
hdisk59	fcs12	
hdisk60	fcs13	D1F06V4ITEM, D1F06V4WARE, D1F06V4DIST, D1F06V4CSTI, D1F06V4NORA, D1F06V4ORL, D1F06V4STK, D1F06V4CST, D1F06V4ORDI, D1F06V4ORD, D1F06V4HIST, D1F06V4NORB
hdisk61	fcs12	
hdisk74	fcs18	D1F07V1ITEM, D1F07V1WARE, D1F07V1DIST, D1F07V1CSTI, D1F07V1NORA, D1F07V1ORL, D1F07V1STK, D1F07V1CST, D1F07V1ORDI, D1F07V1ORD, D1F07V1HIST, D1F07V1NORB
hdisk75	fcs19	
hdisk76	fcs18	D1F07V2ITEM, D1F07V2WARE, D1F07V2DIST, D1F07V2CSTI, D1F07V2NORA, D1F07V2ORL, D1F07V2STK, D1F07V2CST, D1F07V2ORDI, D1F07V2ORD, D1F07V2HIST, D1F07V2NORB
hdisk77	fcs19	
hdisk78	fcs18	D1F07V3ITEM, D1F07V3WARE, D1F07V3DIST, D1F07V3CSTI, D1F07V3NORA, D1F07V3ORL, D1F07V3STK, D1F07V3CST, D1F07V3ORDI, D1F07V3ORD, D1F07V3HIST, D1F07V3NORB
hdisk79	fcs19	
hdisk80	fcs18	D1F07V4ITEM, D1F07V4WARE, D1F07V4DIST, D1F07V4CSTI, D1F07V4NORA, D1F07V4ORL, D1F07V4STK, D1F07V4CST, D1F07V4ORDI, D1F07V4ORD, D1F07V4HIST, D1F07V4NORB
hdisk81	fcs19	
hdisk82	fcs20	D1F08V1ITEM, D1F08V1WARE, D1F08V1DIST, D1F08V1CSTI, D1F08V1NORA, D1F08V1ORL, D1F08V1STK, D1F08V1CST, D1F08V1ORDI, D1F08V1ORD, D1F08V1HIST, D1F08V1NORB
hdisk83	fcs21	
hdisk84	fcs20	D1F08V2ITEM, D1F08V2WARE, D1F08V2DIST, D1F08V2CSTI, D1F08V2NORA,

hdisk85	fcs21	D1F08V2ORL, D1F08V2STK, D1F08V2CST, D1F08V2ORDI, D1F08V2ORD, D1F08V2HIST, D1F08V2NORB
hdisk86	fcs20	D1F08V3ITEM, D1F08V3WARE, D1F08V3DIST, D1F08V3CSTI, D1F08V3NORA, D1F08V3ORL, D1F08V3STK, D1F08V3CST, D1F08V3ORDI, D1F08V3ORD, D1F08V3HIST, D1F08V3NORB
hdisk87	fcs21	
hdisk88	fcs20	D1F08V4ITEM, D1F08V4WARE, D1F08V4DIST, D1F08V4CSTI, D1F08V4NORA, D1F08V4ORL, D1F08V4STK, D1F08V4CST, D1F08V4ORDI, D1F08V4ORD, D1F08V4HIST, D1F08V4NORB
hdisk89	fcs21	
hdisk6	fcs1	D1F09V1ITEM, D1F09V1WARE, D1F09V1DIST, D1F09V1CSTI, D1F09V1NORA, D1F09V1ORL, D1F09V1STK, D1F09V1CST, D1F09V1ORDI, D1F09V1ORD, D1F09V1HIST, D1F09V1NORB
hdisk7	fcs0	
hdisk8	fcs1	D1F09V2ITEM, D1F09V2WARE, D1F09V2DIST, D1F09V2CSTI, D1F09V2NORA, D1F09V2ORL, D1F09V2STK, D1F09V2CST, D1F09V2ORDI, D1F09V2ORD, D1F09V2HIST, D1F09V2NORB
hdisk9	fcs0	
hdisk10	fcs1	D1F09V3ITEM, D1F09V3WARE, D1F09V3DIST, D1F09V3CSTI, D1F09V3NORA, D1F09V3ORL, D1F09V3STK, D1F09V3CST, D1F09V3ORDI, D1F09V3ORD, D1F09V3HIST, D1F09V3NORB
hdisk11	fcs0	
hdisk12	fcs1	D1F09V4ITEM, D1F09V4WARE, D1F09V4DIST, D1F09V4CSTI, D1F09V4NORA, D1F09V4ORL, D1F09V4STK, D1F09V4CST, D1F09V4ORDI, D1F09V4ORD, D1F09V4HIST, D1F09V4NORB
hdisk13	fcs0	
hdisk70	fcs17	D1F10V1ITEM, D1F10V1WARE, D1F10V1DIST, D1F10V1CSTI, D1F10V1NORA, D1F10V1ORL, D1F10V1STK, D1F10V1CST, D1F10V1ORDI, D1F10V1ORD, D1F10V1HIST, D1F10V1NORB
hdisk66	fcs16	
hdisk71	fcs17	D1F10V2ITEM, D1F10V2WARE, D1F10V2DIST, D1F10V2CSTI, D1F10V2NORA, D1F10V2ORL, D1F10V2STK, D1F10V2CST, D1F10V2ORDI, D1F10V2ORD, D1F10V2HIST, D1F10V2NORB
hdisk67	fcs16	
hdisk72	fcs17	D1F10V3ITEM, D1F10V3WARE, D1F10V3DIST, D1F10V3CSTI, D1F10V3NORA, D1F10V3ORL, D1F10V3STK, D1F10V3CST, D1F10V3ORDI, D1F10V3ORD, D1F10V3HIST, D1F10V3NORB
hdisk68	fcs16	
hdisk73	fcs17	D1F10V4ITEM, D1F10V4WARE, D1F10V4DIST, D1F10V4CSTI, D1F10V4NORA, D1F10V4ORL, D1F10V4STK, D1F10V4CST, D1F10V4ORDI, D1F10V4ORD, D1F10V4HIST, D1F10V4NORB
hdisk69	fcs16	

Table 4-2: IBM eServer p5 570 Model 9117-570 Data Distribution Benchmark Configuration

4.5. 60-Day Space Calculations

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

60-Day Space Computation					
All data sizes in MB unless otherwise stated					
Warehouses	64,000				
Measured TpmC	809,144				
Table	Rows	Table	Index	5% Space	Total Space
Warehouse	64,000	11	0	1	12
District	640,000	80	0	4	84
Item	100,000	10	0	1	11
Stock	6,400,000,000	2,083,520	0	104,176	2,187,696
Customer	1,920,000,000	1,500,200	92,720	79,646	1,672,566
New-Order	576,000,000	44,256	0	0	44,256
Orders	1,920,000,000	70,727	54,080	0	124,807
Order-Line	28,800,000,000	1,891,592	0	0	1,891,592
History	1,920,000,000	119,552	0	0	119,552
Free Space	391,719	<u>30 Minute log Computations</u>			
Dynamic Space	2,081,870	Log Written (KB)			56,816,820
Static Space	3,958,704	Total New-Order Txns			24,015,000
Daily Growth	421,133	Log Written per New-Order (KB)			2.37
Daily Spread	0				
Data Storage Requirement					
60 Days (MB)	29,226,698				
60 Days (GB)	28,542				
Log Storage Requirement					
8 Hours (GB)	876.32				
Disk Sizing					
	Formatted	SUT		Priced	
Disk Type	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks	Capacity (GB)
DB FastT 36.4GB	33.40	1,600	53,440	1,600	53,440
LOG FastT RAID10 (40 x 73.4GB)	33.93	40	1,357	40	1,357
OS SCSI 36.4GB	33.88	3	102	3	102
OS SCSI 73.4GB	67.76	2	136	2	136
Total Capacity					54,899

5 Clause 5: Performance Metrics and Response Time Related Items

5.1. Response Times

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

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

5.2. Keying and Think Times

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

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

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	0.26	0.26	0.27	0.14/0.23	0.33	0.15
Average	0.19	0.19	0.19	0.13/0.15	0.22	0.12
Maximum	5.66	5.72	5.63	5.14/4.72	5.80	5.18
Think Times						
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.02	10.01	5.02	5.02	N/A
Maximum	120.25	120.30	100.15	50.23	50.23	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	19.05	3.91	2.76	2.49	2.93	N/A

Table 5-1: Think and Keying Times

5.3. Response Time Frequency Distribution

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

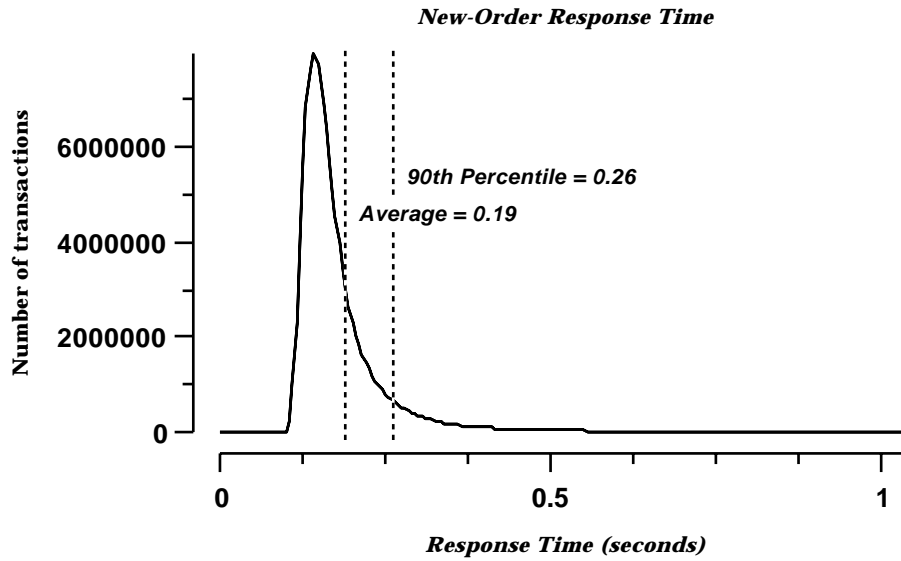


Figure 5-1: New-Order Response Time Distribution

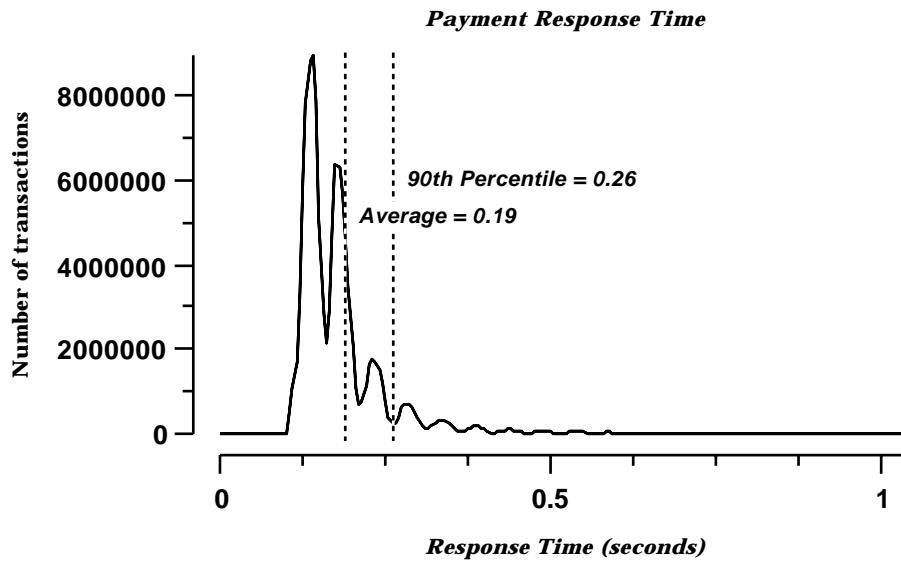


Figure 5-2: Payment Response Time Distribution

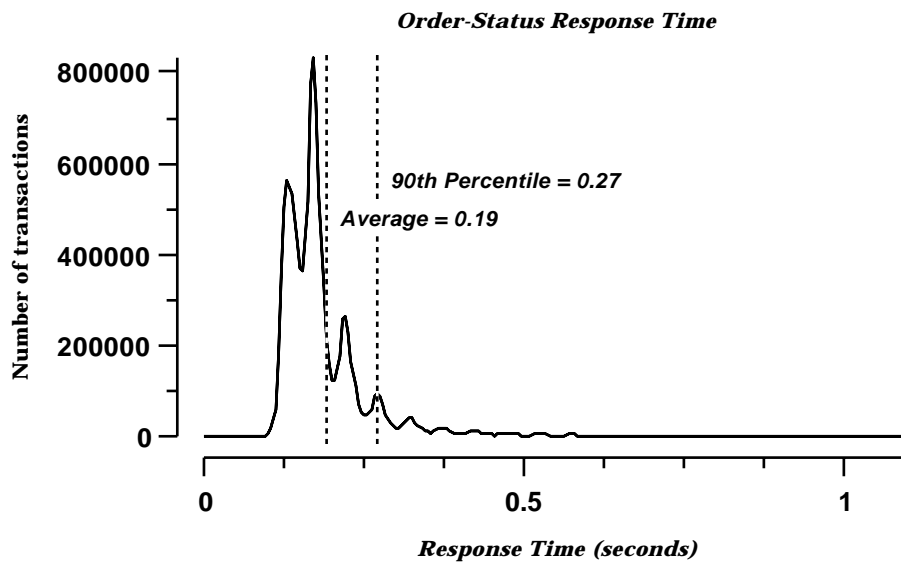


Figure 5-3: Order-Status Response Time Distribution

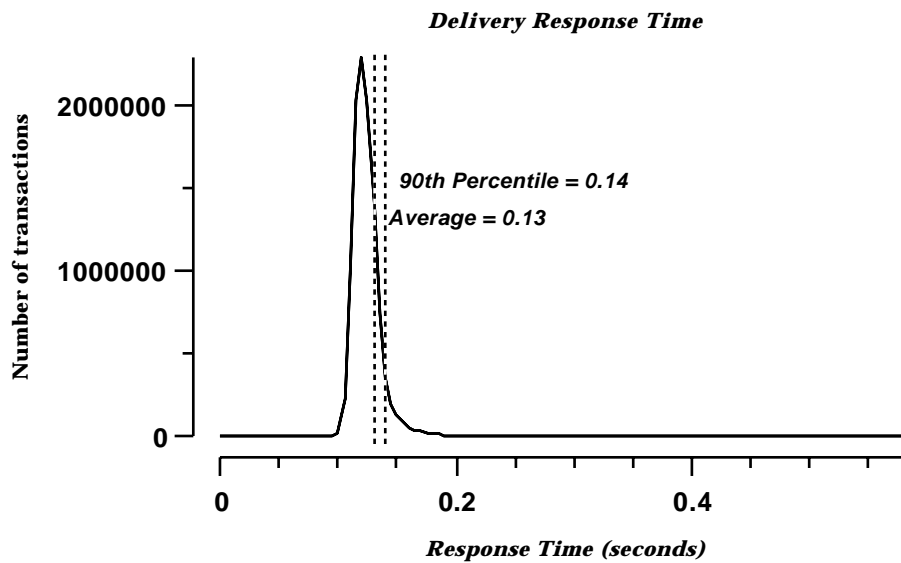


Figure 5-4: Delivery (Interactive) Response Time Distribution

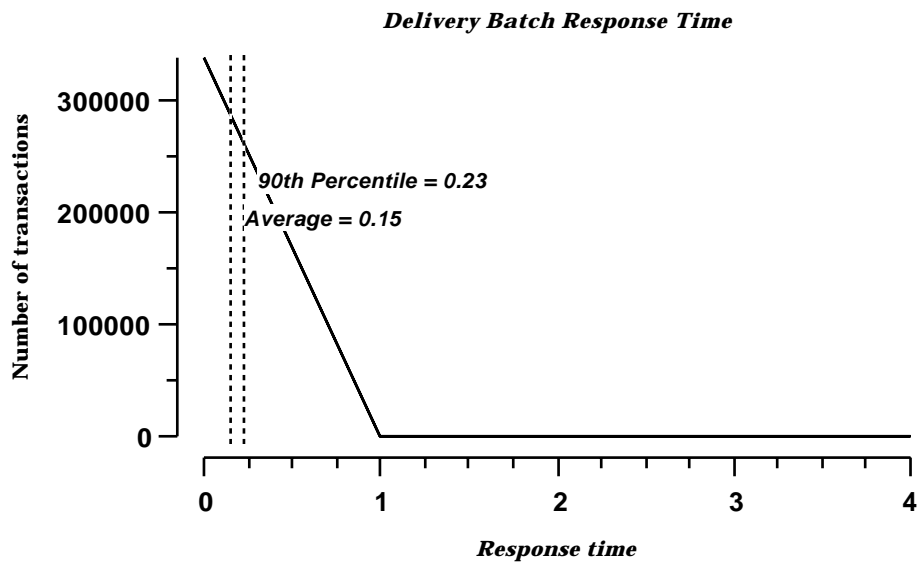


Figure 5-5: Delivery (Deferred) Response Time Distributio

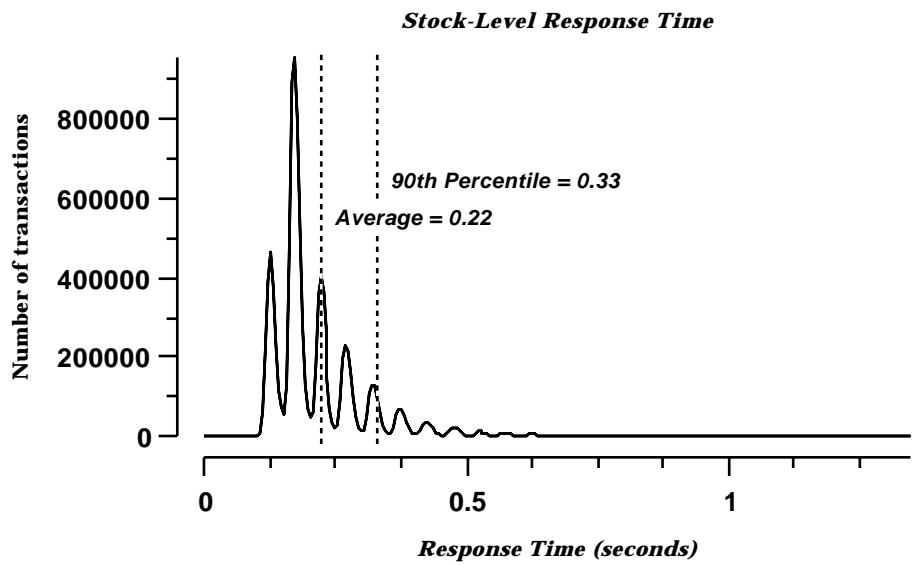


Figure 5-6: Stock Level Response Time Distribution

5.4. Performance Curve for Response Time versus Throughput

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

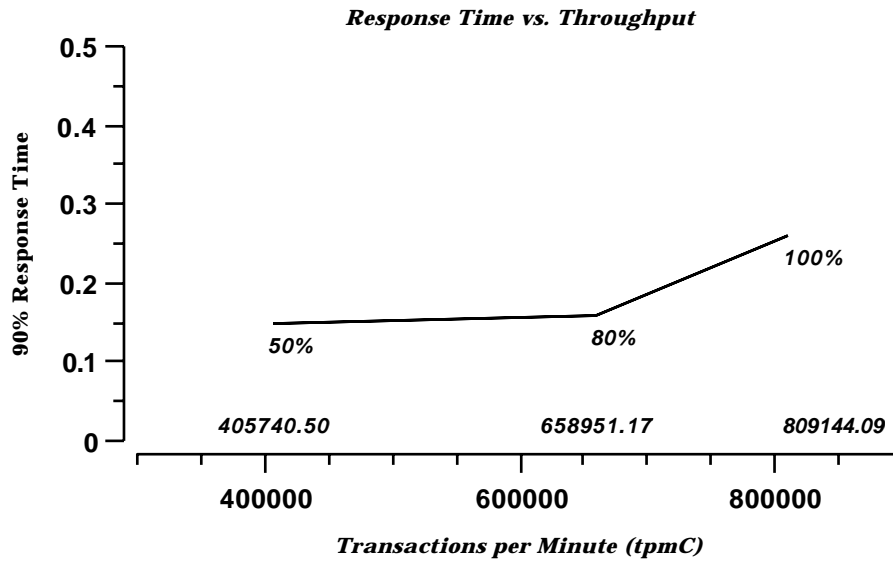


Figure 5-7: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

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

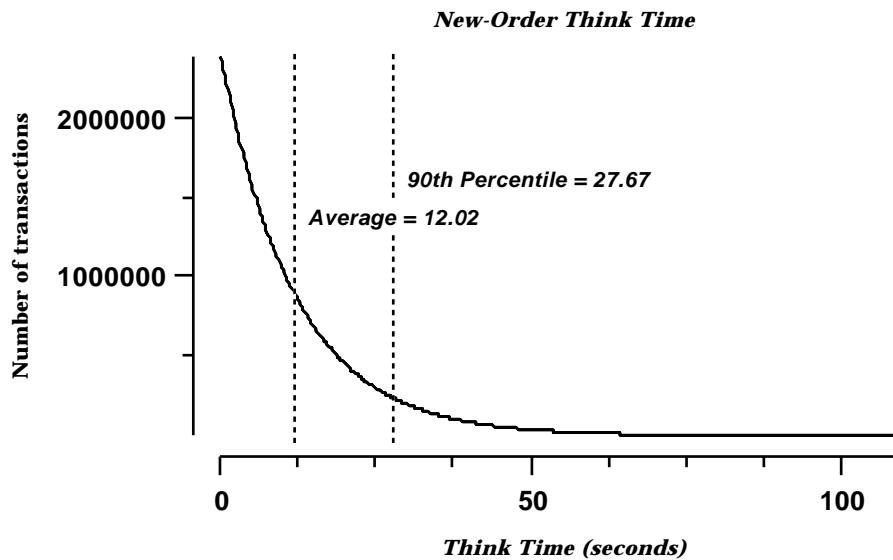


Figure 5-8: New-Order Think Time Distribution

5.6. Throughput versus Elapsed Time

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

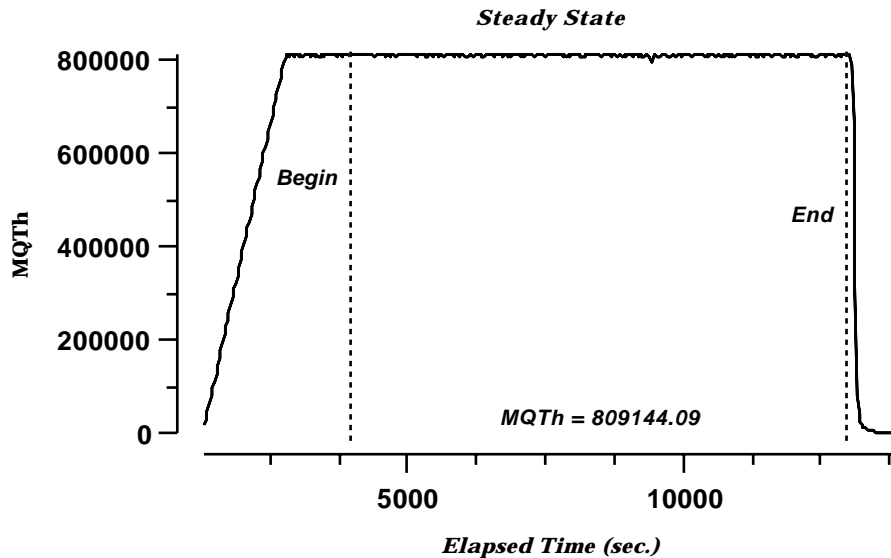


Figure 5-9: New-Order Throughput vs. Elapsed Time

5.7. Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be described.

All the emulated users were allowed to logon and do transactions. The user ramp-up phase is clearly visible on the graph above. Refer to the Numerical Quantities Summary pages for the rampup time. Figure 5-9 New-Order throughput versus Elapsed Time graph shows that the system maintained a steady state during the measurement interval

5.8. Work Performed During Steady State

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

A 2-hour 30-minute measurement interval was used to guaranty that all work normally performed during an 8-hour sustained test are included in the reported throughput.

5.8.1. Transaction Flow

Each of the 4 (non-delivery) transactions is serviced by 2 individual programs, Internet Information System 5.1 (IIS) and a Microsoft COM+ 1.0 Queued Component Server, used as the transaction manager (COM+). Both programs are running on the client system:

- The initial HTML 1.0 request is serviced by an ISAPI custom-written handler running on Internet Information System 5.1(IIS). IIS is responsible for handling all HTML requests. The web server communicates to the COM+ server through a Microsoft COM+ api interface.
- COM+ communicates with the Server system over Ethernet and handles all database operations, using DB2 embedded SQL calls.
When the COM+ server boots up, it creates a configurable amount of connections to the Server (listed in application settings).

COM+ routes the transaction and balances the load according to the options defined in the Component Services GUI for the COM+ server application and settings in the Windows 2000 Registry. The configuration file and registry variables are listed in Appendix B.2.

At the beginning, each TPC-C user sends a pair of HTML 1.0 requests submitting the its unique warehouse and district to the IIS ISAPI handler. Upon successful validation of user's login, IIS the displays an HTML form which encapsulates the TPC-C transaction menu.

The transaction flow is described below:

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

5.8.2. Database Transaction

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

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

DB2 UDB Server proceeds to update the database as follows:

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

5.8.3. Checkpoints

DB2 UDB uses a write-ahead-logging protocol to guarantee recovery. This protocol uses "Soft" checkpoint to write least-recently-used database pages to disk independent of transaction commit. However, enough log information to redo/undo the change to a database pages is committed to disk before the database page itself is written. This protocol therefore renders checkpoint unnecessary for DB2 UDB. For a more detailed description of the general principles of the write-ahead-logging protocol, see the IBM research paper, "ARIES: A Transaction Recovery Method Supporting Fine Granularity Locking and Partial Rollbacks Using Write-Ahead Logging," by C. Mohan, Database Technology Institute, IBM Almaden Research Center.

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

5.9. Measurement Interval

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

A 2-hour 30-minute measurement interval was used. No connections were lost during the run.

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

6.1. RTE Availability

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs to the RTE had been used.

IBM used an internally developed RTE for these tests. Appendix D contains the scripts used in the testing.

6.2. Functionality and Performance of Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system.

No components were emulated.

6.3. Network Bandwidth

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

The 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 with a 1Gigabit ethernet uplink which is connected to the database server. The Gigabit Ethernet line connecting the clients and the database server has a bandwidth of 1000 Mbits per second.

6.4. Operator Intervention

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

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

7 Clause 7: Pricing Related Items

7.1. Hardware and Programs Used

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

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

7.2. Three Year Cost of System Configuration

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

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

Discount are based on US list prices and for similar quantities and configurations. A discount of 39.5% has been applied to all IBM hardware, software, and services based on the total value and quantities of the components of the configuration, including full payment of all components, including prepaid maintenance for three years.

7.3. Availability Dates

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

All products are generally available today except the following:

Product	Availability Date
IBM eServer p5 570 Model 9117-570	September 30, 2004
AIX 5L Version 5.3	August 31, 2004
DB2 Universal Database 8.1	August 16, 2004

7.4. Statement of tpmC and Price/Performance

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

.System	tpmC	3-year System Cost	\$/tpmC	Availability Date
IBM eServer p5 570 Model 9117-570	809,144.09	\$4,004,491 USD	\$4.95 USD	September 30, 2004

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

8 Clause 9: Audit Related Items

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

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

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

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

June 23, 2004

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

Platform: IBM eServer p5 570 Model 9117-570 c/s
 Operating system: AIX 5L V5.3
 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 p5 570 Model 9117-570				
16 x POWER5 (1.9 GHz)	512 GB main (36 MB L3 cache/DCM)	3 x 36.4GB SCSI int. 2 x 73.4GB SCSI int. 1600 x 36.4GB FASTT 40 x 73.4GB FASTT	0.26 Sec.	809,144.09
Thirty-two (32) Clients: IBM eServer xSeries 335 Model 8676-G2X (Specification for each)				
2 x Intel Xeon (3.2 GHz)	2 GB main (1 MB L2 cache/cpu)	1 x 80GB	n/a	n/a

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

The following verification items were given special attention:

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

- The ACID properties were met.
- Input data was generated according to the specified percentages.
- The transaction cycle times included the required browser delay, keying and think times.
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit.
- All 90% response times were under the specified maximums.
- The measurement interval was representative of steady state conditions.
- The reported measurement interval was 2.5 hours.
- Write-ahead-logging was active during the measurement interval.
- The 60 day storage requirement was correctly computed.
- The system pricing was verified for major components and maintenance.

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink, appearing to read "François Raab". The signature is fluid and cursive, with a long horizontal stroke extending to the right.

François Raab, President

Appendix - A: Client Server Code

A.1 Client/Terminal Handler Code

Makefile.config

```
#####  
#####  
## Licensed Materials - Property of IBM  
##  
## Governed under the terms of the International  
## License Agreement for Non-Warranted Sample Code.  
##  
## (C) COPYRIGHT International Business Machines Corp. 1996 -  
2004  
## All Rights Reserved.  
##  
## US Government Users Restricted Rights - Use, duplication or  
## disclosure restricted by GSA ADP Schedule Contract with IBM  
Corp.  
#####  
#####  
  
#  
# Makefile.config - NT/Win2000 Makefile Configuration  
#  
  
# Make Configuration (MSVC)  
MAKE=nmake.exe  
  
# Compiler Configuration (MSVC).  
CC=cl.exe  
CFLAGS_OS=-DSQLWINT -MT -DWIN32 -J -Zp8 -  
DREG_KIT_METHOD -DSWAP_ENDIAN  
CFLAGS_OUT=/Fo  
CFLAGS_DEBUG=  
  
# Linker Configuration (MSVC)  
LD_EXEC=link.exe  
LD_STORP=link.exe  
LDFLAGS_EXEC=  
LDFLAGS_SHLIB=/DLL  
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpcc.def  
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib  
/LIBPATH:"C:\Program Files\Microsoft Visual Studio\VC98\Lib"  
db2api.lib winmm.lib  
LDFLAGS_OUT=/OUT:  
  
# Library Configuration  
AR=lib.exe  
ARFLAGS=  
ARFLAGS_LIB=  
ARFLAGS_OUT=/OUT:  
  
# OS Commands  
ERASE=del /F  
ERASEDIR=rmdir /S  
MOVE=MOVE  
COPY=COPY  
  
# OS File Extensions & Path Separator  
OBJEXT=.obj  
LIBEXT=.lib  
SHLIBEXT=.dll  
BINEXT=.exe  
SLASH=\  
CMDSEP=&
```

tpccenv.bat

```
@REM  
*****  
@REM Licensed Materials - Property of IBM  
@REM  
@REM Governed under the terms of the International  
@REM License Agreement for Non-Warranted Sample Code.  
@REM  
@REM (C) COPYRIGHT International Business Machines Corp.  
1996 - 2004  
@REM All Rights Reserved.  
@REM  
@REM US Government Users Restricted Rights - Use, duplication  
or  
@REM disclosure restricted by GSA ADP Schedule Contract with  
IBM Corp.  
@REM  
*****  
  
@REM  
@REM tpccenv.bat - Windows Environment Setup  
@REM  
  
@REM The Kit Version  
set TPCC_VERSION=CK040318  
  
@REM The DB2 Instance Name (for DB2)  
set DB2INSTANCE=%USERNAME%  
  
@REM The OS being used (i.e. "UNIX", "WINDOWS")  
set PLATFORM=WINDOWS  
  
@REM The type of make command and slash used by the OS  
@REM (i.e. UNIX - "/", WINDOWS - "\")  
@REM These are referenced all over the kit.  
set SLASH=\  
set MAKE=nmake  
  
@REM Specifies whether or not to use dari stored proc's for the  
TPC-C driver. Set to either DARIVERSION or NONDARI;  
@REM set TPCC_SPTYPE=NOSP  
@REM set TPCC_SPTYPE=SPGENERAL2  
set TPCC_SPTYPE=SPGENERAL  
@REM set TPCC_SPTYPE=DARI2SQLDA  
  
set DB2VERSION=v8  
  
@REM The schema name is typically the SQL authorization ID (or  
username).  
@REM This is required for runstats and EEE.  
set TPCC_SCHEMA=%USERNAME%  
  
@REM DB2 EE/EEE Configuration  
set DB2EDITION=EE  
@REM set DB2EDITION=EEE  
set DB2NODE=0  
@REM set to the number of nodes you have. Set to 1 for EE.  
set DB2NODES=1  
  
@REM TPCC General Configuration  
@REM ** IMPORTANT NOTE **  
@REM The kit is not guaranteed to work properly if TPCC_ROOT or  
TPCC_SQLLIB  
@REM have spaces in them. If you absolutely must use paths with  
spaces,  
@REM then the entire path must be surrounded by double quotes.  
@REM For example: HOME="C:\Program Files\IBM"  
set HOME=C:\home\tpcc  
set TPCC_DBNAME=TPCC  
set TPCC_ROOT=%HOME%\tpc-c.ibm
```

```

set TPCC_SQLLIB=C:\Progra-1\IBM\sqllib
set TPCC_RUNDATA=%HOME%\tpc-c.ibm\tpccdata

@REM TPCC Debug Configuration
@REM This is the path where all error and debug logs are placed.
@REM To get debugging from within the stored procedures, you
must
@REM set DB2ENVLIST="TPCC_DEBUGDIR" in tpc.config.
set TPCC_DEBUGDIR=c:\temp

@REM Specifies where stored procedures should be placed and if
they should
@REM be fenced.
set TPCC_SPDIR=%TPCC_SQLLIB%\function
set TPCC_FENCED=NO

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>
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_ol_cnt;
    int16_t pad1[2];
    struct oitems_struct {
        int64_t s_OL_DELIVERY_D_time;
        int32_t s_OL_AMOUNT;
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
    };
};

```

```

    int16_t s_OL_QUANTITY;
    int16_t pad2;
} item[15];
int16_t s_transtatus;
int16_t deadlocks;
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
};

```

```

struct in_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_O_DELIVERY_D_time; /* init by SUT */
    int32_t s_W_ID;
    int16_t s_O_CARRIER_ID;
};

```

```

struct out_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_O_ID[10];
    int16_t s_transtatus;
    int16_t deadlocks;
};

```

```

struct in_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_threshold;
    int32_t s_W_ID;
    int16_t s_D_ID;
};

```

```

struct out_stocklev_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_low_stock;
    int16_t s_transtatus;
    int16_t deadlocks;
};

```

```

/* ***** */
/* Transaction Prototypes */
/* ***** */

```

```

#ifdef __cplusplus
extern "C" {
#endif

```

```

extern int neword_sql(struct in_neword_struct*, struct
out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct
out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct
out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct
out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct
out_stocklev_struct*);

```

```

#ifdef __cplusplus
}
#endif

```

```

/* ***** */
/* DB2 Connect/Disconnect & Thread Context Wrappers */
/* ***** */

```

```

#ifdef __cplusplus
extern "C" {

```



```

#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

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

#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H

include/lval.h

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 64000
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

include/tpccapp.h

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

/*
 * tpccapp.h - Application Macros
 */

#ifndef __TPCCAPP_H
#define __TPCCAPP_H

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

#include "sqlenv.h"
#define daricall __stdcall

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

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

/*****
*****

```

```

FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Addr + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78

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

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

        *a ^= *b;
        *b ^= *a;
        *a ^= *b;
    }
}

#endif //SWAP_ENDIAN

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

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

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

#define NACOMPCHK(last) \
    if (sqlca.sqlcode != SQL_RC_E139) { last = -1; } \
    else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : sqlca.sqlerrmc[4]-0x30); \
           int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-0x30); \
           if (b == 0) { last = a; } else { last = a * 10 + b; } \
    }

#endif // __TPCCAPP_H

```

include/tpccdbg.h

```

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

/*
 * tpccdbg.h - Debugging Macros
 */

#ifndef __TPCCDBG_H
#define __TPCCDBG_H

#ifdef __cplusplus
extern "C" {
#endif

extern void new_debug (struct out_neword_struct *neword_ptr,
                      struct in_neword_struct *in_neword_ptr,
                      char *msg);
extern void pay_debug (struct out_payment_struct *payment_ptr,
                      struct in_payment_struct *in_payment_ptr,
                      char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                      struct in_ordstat_struct *in_ordstat_ptr,
                      char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
                      struct in_delivery_struct *in_delivery_ptr,
                      char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
                      struct in_stocklev_struct *in_stocklev_ptr,
                      char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
                      struct in_neword_struct *in_neword_ptr,
                      char *filename,
                      char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
                      struct in_payment_struct *in_payment_ptr,
                      char *filename,
                      char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
                      struct in_ordstat_struct *in_ordstat_ptr,
                      char *filename,
                      char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
                      struct in_delivery_struct *in_delivery_ptr,
                      char *filename,
                      char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
                      struct in_stocklev_struct *in_stocklev_ptr,
                      char *filename,
                      char *msg);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H

```

Src.Common/Makefile

```

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

#
# Makefile - Makefile for Src.Common
#

!include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####

BND_OPTS = GRANT PUBLIC \
           MESSAGES $*.bnd.msg
PRP_OPTS = BINDFILE \
           OPTLEVEL 1 \
           ISOLATION RR \
           MESSAGES $*.prep.msg \
           LEVEL $(TPCC_VERSION) \
           NOLINEMACRO

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

CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \
         -DSQLA_NOLINES -D$(DB2EDITION) -D$(DB2VERSION) \
         -D$(TPCC_SPTYPE)

UTIL_OBJ = tpccdbg$(OBJEXT) tpcctx$(OBJEXT)

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

all: connect $(UTIL_OBJ) disconnect

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

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

connect:

```

```
- db2 connect to $(TPCC_DBNAME)
```

```
disconnect:
```

```
- db2 connect reset  
- db2 terminate
```

```
rebind:
```

```
db2 bind tpcctx.bnd $(BND_OPTS)
```

```
#  
#####  
#####  
# Build Rules  
#  
#####  
#####
```

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

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

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

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

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

Src.Common/tpcctx.sqc

```
/*  
*****  
*****  
** Licensed Materials - Property of IBM  
**  
** Governed under the terms of the International  
** License Agreement for Non-Warranted Sample Code.  
**  
** (C) COPYRIGHT International Business Machines Corp. 1996 -  
2004  
** All Rights Reserved.  
**  
** US Government Users Restricted Rights - Use, duplication or  
** disclosure restricted by GSA ADP Schedule Contract with IBM  
Corp.  
*****/  
  
/*  
* tpcctx.sqc - TPCC context code  
*/  
  
#include <stdlib.h>  
#include <stdio.h>  
#include <sqlutil.h>  
#include "db2tpcc.h"
```

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

```
int connect_to_TM(char *in_dbname)  
{  
    return connect_to_TM_auth(in_dbname, "", "");  
}
```

```
int connect_to_TM_auth(char *in_dbname, char *in_username, char  
*in_password)  
{  
    SQL_STRUCTURE sqlca sqlca;  
    int ConnectSQLCODE = 0;
```

```
EXEC SQL BEGIN DECLARE SECTION;  
char dbname[9];  
char username[129];  
char password[15];  
EXEC SQL END DECLARE SECTION;
```

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

```
strncpy(dbname,in_dbname,8);  
if (strcmp(in_username,"") == 0)  
{  
    EXEC SQL CONNECT TO :dbname IN SHARE MODE;  
} else {  
    strncpy(username,in_username,128);  
    strncpy(password,in_password,14);  
    EXEC SQL CONNECT TO :dbname IN SHARE MODE  
USER :username USING :password;  
}
```

```
ConnectSQLCODE = SQLCODE;  
if (ConnectSQLCODE != 0)  
{  
    sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__,  
&sqlca);
```

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

```
return ConnectSQLCODE;  
}
```

```
return 0;
```

```
int disconnect_from_TM(void)
```

```
{  
    SQL_STRUCTURE sqlca sqlca;  
    int DisconnectSQLCODE = 0;
```

```
EXEC SQL CONNECT RESET;
```

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

```
SQLCODE = destroy_context();
```

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

```

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

int create_context(void)
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;

    sqleSetTypeCtx(SQL_CTX_MULTI_MANUAL);
    sqleBeginCtx(&ctx, SQL_CTX_BEGIN_ALL, NULL, &sqlca);

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

    return 0;
}

int attach_context(void *ctx)
{
    SQL_STRUCTURE sqlca sqlca;

    sqleAttachToCtx(ctx, NULL, &sqlca);

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

    return 0;
}

int detach_context(void *ctx)
{
    SQL_STRUCTURE sqlca sqlca;

    sqleDetachFromCtx(ctx, NULL, &sqlca);

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

    return 0;
}

int destroy_context(void)
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;

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

    sqleEndCtx(&ctx, SQL_CTX_END_ALL, NULL, &sqlca);

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

    return 0;
}

```

```

}

int get_context(void **ctx)
{
    SQL_STRUCTURE sqlca sqlca;

    sqleGetCurrentCtx(ctx, NULL, &sqlca);

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

    return 0;
}

```

Src.Common/tpccdbg.c

```

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

/*
 * tccdbg.c - Debugging Routines
 */

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

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

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

void del_print();
void new_print();
void ord_print();
void pay_print();
void stk_print();

void current_tmstamp(char *buf);

static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";

/*-----*/
/* InitializeDebug */
/*-----*/
__inline void InitializeDebug(void) {
    if (debugInit == 0) {
        char *p = getenv("TPCC_DEBUGDIR");
        if (p) {
            strncpy(debugPath, p, DEBUG_PATH_SIZE);

```

```

    } else {
        strcpy(debugPath, "C:\\temp");
    }
    strcat(debugPath, "\\");
}
debugInit = 1;
}

/*-----*/
/*      sqlerror      */
/*-----*/
void sqlerror(int tranType, char *msg, char *file, int line,
SQL_STRUCTURE sqlca *psqlca)
{
    FILE *err_fp = NULL;
    char err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
    char tranName[16];
    int j,k;
    char timeStamp[27];
    char errStr[512] = "";

    InitializeDebug();
    strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
    current_tmstamp(&timeStamp[0]);
    timeStamp[19] = (char)NULL;

    switch(tranType)
    {
        case NEWORD_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "new.err.out");
            strcpy(tranName, "NEW_ORDER");
            break;

        case DELIVERY_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "del.err.out");
            strcpy(tranName, "DELIVERY");
            break;

        case PAYMENT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "pay.err.out");
            strcpy(tranName, "PAYMENT");
            break;

        case ORDSTAT_SQL:
            // sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "ord.err.out");
            strcpy(tranName, "ORDER_STAT");
            break;

        case STOCKLEV_SQL:
            //sprintf(err_fn, "%d.err.out", getpid());
            strcat(err_fn, "stk.err.out");
            strcpy(tranName, "STOCK_LVL");
            break;

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

        default:
            return;
    }

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

    err_fp = fopen(err_fn, "a+");

```

```

fprintf(err_fp, "-----\n");
fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
fprintf(err_fp, "FILE %s (%u)\n", file, line);
fprintf(err_fp, "SQLCODE %d", psqlca->sqlcode);
fprintf(err_fp, "TIME %s\n", timeStamp);
fprintf(err_fp, "-----\n");
fprintf(err_fp, "%s", errStr);
fprintf(err_fp, "-----\n");

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

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

fprintf(err_fp, "sqlerrp: ");
for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
fprintf(err_fp, "\n");

fprintf(err_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
    fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
fprintf(err_fp, "\n");

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

fprintf(err_fp, "\n");

fclose(err_fp);
}

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

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "del.debug.out");

```



```

neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "\ts_O_ID = %d (%X)\n",
neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp, "\ts_O_OL_CNT = %d (%X)\n",
neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
fprintf(debug_fp, "\ts_O_ENTRY_D = %lld (%lX)\n",
neword_ptr->s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);
fprintf(debug_fp, "\ts_total_amount = %d\n",
neword_ptr->s_total_amount);
fprintf(debug_fp, "\ts_transtatus = %d (%X)\n",
neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "\tdeadlocks = %d (%X)\n",
neword_ptr->deadlocks, neword_ptr->deadlocks);

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

fprintf(debug_fp, "\titems {\n");
items = neword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "\ts_I_NAME[%d] = %s\n",
        j, neword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp, "\ts_I_PRICE[%d] = %d\n",
        j, neword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp, "\ts_OL_AMOUNT[%d] = %d\n",
        j, neword_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "\ts_S_QUANTITY[%d] = %d (%X)\n",
        j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr-
>item[j].s_S_QUANTITY);
    fprintf(debug_fp, "\ts_brand_generic[%d] = %c\n",
        j, neword_ptr->item[j].s_brand_generic);
}
fprintf(debug_fp, "\t}\n\n");
fclose(debug_fp);
}

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

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

/*-----*/
/* ord_print */
/*-----*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct *in_ordstat,
                char *filename,
                char *msg)
{
    FILE *debug_fp;

```

```

char timeStamp[27];
int j, items;

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

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

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

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

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

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

```

Src.Cli/Makefile

```

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

```

```

## disclosure restricted by GSA ADP Schedule Contract with IBM
Corp.
#####
#####

#
# Makefile - Makefile for Src.Cli (RTE/Driver Interface)
#

!include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
           MESSAGES $*.bnd.msg
PRP_OPTS = BINDFILE \
           ISOLATION RR \
           EXPLAIN ALL \
           MESSAGES $*.prep.msg \
           LEVEL $(TPCC_VERSION) \
           NOLINEMACRO

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

CFLAGS = $(CFLAGS_OS) $(INCLUDES) $(CFLAGS_DEBUG)
\
$(UOPTS) -D$(DB2EDITION) -D$(DB2VERSION) -
D$(TPCC_SPTYPE)

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

LIBS = tpcccli$(LIBEXT)

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

all: connect $(OBS) plan $(LIBS) disconnect
$(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT)
$(OBS) $(ARFLAGS_LIB)
@echo "-----"
@echo "Please copy lval.h, db2tpcc.h, and tpcccli$(LIBEXT)
to"
@echo "a place where they can be #included and linked with
the"
@echo "RTE/driver code."
@echo "-----"

clean:
- $(ERASE) *.msg *.bnd *.plan *$(OBJEXT) *$(LIBEXT)
tpcccli.c

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

```

```

#
#####
#####

connect:
  - db2 connect to $(TPCC_DBNAME)

disconnect:
  - db2 connect reset
  - db2 terminate

plan:
  - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s
$(TPCC_SCHEMA) -w -1 -n TPCCCLI -g -# 0 -o
TPCCCLI.exfmt.plan
  - db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p
TPCCCLI -s 0 -g -o TPCCCLI.expln.plan

rebind: connect
      db2 bind tpcccli.bnd $(BND_OPTS) QUERYOPT 7

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

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

tpcccli.c:
  @echo "Prepping $*.sqc"
  -db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
  @echo "Binding $*.bnd"
  db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7

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

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

# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c

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

Src.Cli/tpcccli.sqc

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

```

```

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

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

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

#include "lval.h"

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

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

  // If diff item id then sort on that.
  // If real/quasi dup, then sort on warehouse id.

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

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

  EXEC SQL BEGIN DECLARE SECTION;

  struct vc_new_in
  {
    short len;
    char data[ 270 ] ;
  } * pHostvarInput ;

  struct vc_new_out
  {
    short len;
    char data[ 662 ] ;
  } * pHostvarOutput ;

  EXEC SQL END DECLARE SECTION;

  int clientRc = TRAN_OK ;

  int itemIndex = 0 ;
  int actualItemIndex = 0 ;

  /* Create Timestamp */
  in_neword->s_O_ENTRY_D_time = time(NULL) ;

  // Sort the item list ; This helps eliminate duplicates anyway, and
  since invalid item

```

```

// IDs == 100001 , we will remain compliant with 2.4.2.3 Comment
1.
// !! So DON'T sort or change the order of the items in any
subsequent code, including SQL.

in_neword->s_all_local = 1 ;

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

// Pre-duplicate processing count
in_neword->s_O_OL_CNT = itemIndex ;

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

// Now purge the duplicates.

actualItemIndex = -1 ;

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

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

in_neword->s_O_OL_CNT = actualItemIndex + 1 ;

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

pHostvarOutput = (struct vc_new_out *) neword ;

```

```

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

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

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

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

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

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

    if ( sqlca.sqlcode == 0 )
    {
        double wtax = neword->s_W_TAX / 10000.0 ;
        double dtax = neword->s_D_TAX / 10000.0 ;
        double cdisc = neword->s_C_DISCOUNT / 10000.0 ;
        double factor = (1.0 - cdisc) * (1.0 + wtax + dtax) ;
    }

```

```

// Post process the item set, detecting any bad items , and set or
count from that.
// Anything that could be deferred from the SP to the client has
been.

neword->s_total_amount = 0 ;

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

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

    }
}

// s_total_amount gets cast implicitly to a double to do the
arithmetic,
// and then cast back to a sqlint32.
neword->s_total_amount *= factor;
}
else
{
    sqlerror( NEWORD_SQL, "NEW", __FILE__, __LINE__,
    &sqlca );
    neword->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;
}

/* Update Output Structure with Timestamp */
neword->s_O_ENTRY_D_time = in_neword-
>s_O_ENTRY_D_time ;

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

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

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

return ( clientRc ) ;
}

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

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

    int clientRc = TRAN_OK ;

```

```

EXEC SQL BEGIN DECLARE SECTION;

// Inputs

sqlint64 h_amount ;
sqlint32 in_c_id ;

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

sqlint32 w_id ;
sqlint32 c_w_id ;
short d_id ;
short c_d_id ;
sqlint64 h_date;

// Outputs

sqlint32 c_id ;

sqlint64 c_credit_lim ;
sqlint32 c_discount ;
sqlint64 c_balance ;

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

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

char c_last [ 16 ] ;

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

char c_credit [ 2 ] ;

sqlint64 c_since ;

char c_data [ 200 ] ;
short c_data_indicator = 0 ;

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

EXEC SQL END DECLARE SECTION;

// Input redirects

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

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

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

// Output redirects

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

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

```

```

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

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

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

/* Create Timestamp */
in_payment->s_H_DATE_time = (sqlint64) time( NULL ) ;

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

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

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

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

if ( in_c_id == 0 )
{
    // Pre-built c_data prefix for BC does not include c_id in this
    instance

    // 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
                          , :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 ;

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

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

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

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

EXEC SQL ROLLBACK WORK ;

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

return ( clientRc ) ;
}

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

```

```

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

EXEC SQL BEGIN DECLARE SECTION;

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

struct vc_ord_out
{
short len ;
char data[ 446 ] ;
} * out_ord ;

EXEC SQL END DECLARE SECTION;

int clientRc = TRAN_OK ;
int itemIndex = 0 ;

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

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

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

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

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

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

SWAP_BYTE(ordstat->s_C_BALANCE);
SWAP_BYTE(ordstat->s_O_ENTRY_D_time);
SWAP_BYTE(ordstat->s_C_ID);
SWAP_BYTE(ordstat->s_O_ID);
SWAP_BYTE(ordstat->s_O_CARRIER_ID);
SWAP_BYTE(ordstat->s_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( ORDDSTAT_SQL, "ORD", __FILE__, __LINE__,
&sqlca );
ordstat->s_transtatus = FATAL_SQLERROR ;
clientRc = FATAL_SQLERROR ;
}
}

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

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

return ( clientRc ) ;
}

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

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

EXEC SQL BEGIN DECLARE SECTION;

    struct vc_del_in
    {
        short len ;
        char data[ 22 ] ;
    } * in_del ;

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

EXEC SQL END DECLARE SECTION;

int clientRc = TRAN_OK ;
int orderIndex = 0 ;

/* Create Timestamp */

in_delivery->s_O_DELIVERY_D_time = (sqlint64) time( NULL ) ;

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

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

```

```

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

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

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

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

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

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

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

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

return ( clientRc ) ;
}

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

#undef w_id
#undef d_id

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

int clientRc = TRAN_OK ;

EXEC SQL BEGIN DECLARE SECTION;

// input

###sqlint32 w_id ;
###short d_id ;
sqlint32 threshold ;

// output

sqlint32 low_stock ;

```

```

EXEC SQL END DECLARE SECTION;

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

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

#ifdef DEBUGIT
stk_debug(stocklev, in_stocklev, "Client before SQL call");
#endif /* DEBUGIT */

retry_tran:

    stocklev->deadlocks ++ ;

/*
EXEC SQL

    SELECT ITEMS_BELOW_THRESHOLD

        INTO :low_stock

    FROM TABLE( STOCK_LEVEL( :w_id, :d_id, :threshold ) ) AS
T
    WITH CS;
*/

EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

    SELECT COUNT( S_I_ID ) INTO :low_stock

    FROM ( SELECT DISTINCT S_I_ID

        FROM ORDER_LINE , STOCK , DISTRICT

        WHERE D_W_ID = :w_id
            AND D_ID = :d_id
            AND OL_O_ID < d_next_o_id
            AND OL_O_ID >= ( d_next_o_id - 20 )
            AND OL_W_ID = D_W_ID
            AND OL_D_ID = D_ID
            AND S_I_ID = OL_I_ID
            AND S_W_ID = OL_W_ID
            AND S_QUANTITY < :threshold

        ) OLS

    WITH CS
;

COMMIT ;

END COMPOUND ;

#ifdef DEBUGIT
stk_debug(stocklev, in_stocklev, "Client after SQL call");
#endif /* DEBUGIT */

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

    sqlerror( STOCKLEV_SQL , "STK" , __FILE__ , __LINE__ ,
&sqlca);
stocklev->s_transtatus = FATAL_SQLERROR ;
clientRc = FATAL_SQLERROR ;

    stk_debug( stocklev, in_stocklev, "STK failed" ) ;

```

```

EXEC SQL ROLLBACK WORK ;

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

return ( clientRc ) ;
}

```

nullDB/nullDB.h

```

// The following ifdef block is the standard way of creating macros
which make exporting
// from a DLL simpler. All files within this DLL are compiled with the
NULLDB_EXPORTS
// symbol defined on the command line. this symbol should not be
defined on any project
// that uses this DLL. This way any other project whose source files
include this file see
// NULLDB_API functions as being imported from a DLL, whereas
this DLL sees symbols
// defined with this macro as being exported.
#ifdef NULLDB_EXPORTS
#define NULLDB_API __declspec(dllexport)
#else
#define NULLDB_API __declspec(dllimport)
#endif

extern NULLDB_API int dataSet;

extern "C" NULLDB_API int do_nord(struct nord_wrapper *nord,void
*ctx);
extern "C" NULLDB_API int do_pymt(struct paym_wrapper
*pymt,void *ctx);
extern "C" NULLDB_API int do_or ds(struct or ds_wrapper *or ds,void
*ctx);
extern "C" NULLDB_API int do_dlv y(struct dlv y_wrapper *dlv y,void
*ctx);
extern "C" NULLDB_API int do_stok(struct stok_wrapper *stok,void
*ctx);

extern "C" NULLDB_API int connect_db(char *dbName,void **ctx);
extern "C" NULLDB_API int disconnect_db(void *ctx);

```

nullDB/nullDB.cpp

```

// nullDB.cpp : Defines the entry point for the DLL application.
//

#include "stdafx.h"
#include "nullDB.h"
#include "..\tpccsapi\tpcc.h"

BOOL APIENTRY 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");
        strcpy(pymt->out_paym.s_C_STREET_2,"Flat 343");
        strcpy(pymt->out_paym.s_C_CITY,"Cambridge");
        strcpy(pymt->out_paym.s_C_STATE,"NY");
        strcpy(pymt->out_paym.s_C_ZIP,"785585432");
        strcpy(pymt->out_paym.s_C_PHONE,"1234567890123456");
        pymt->out_paym.s_C_SINCE_time = 0;
        strcpy(pymt->out_paym.s_C_CREDIT,"BC");
        strcpy(pymt-
>out_paym.s_C_DATA,"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX");
        dataSet = 1;
    }
}

```

```

else
{
    pymt->out_paym.s_C_CREDIT_LIM = 4000000;
    pymt->out_paym.s_C_DISCOUNT = 52400;
    pymt->out_paym.s_C_BALANCE = 14080;
    pymt->out_paym.s_C_ID = 3180;
    pymt->out_paym.s_H_DATE_time = 1234567890;

    strcpy(pymt->out_paym.s_W_STREET_1,"1201 Park Ave.");
    strcpy(pymt->out_paym.s_W_STREET_2,"Suite 432");
    strcpy(pymt->out_paym.s_W_CITY,"Denver");
    strcpy(pymt->out_paym.s_W_STATE,"CO");
    strcpy(pymt->out_paym.s_W_ZIP,"787562356");
    strcpy(pymt->out_paym.s_D_STREET_1,"3404 Garth Rd");
    strcpy(pymt->out_paym.s_D_STREET_2,"Suite 320");
    strcpy(pymt->out_paym.s_D_CITY,"Austin");
    strcpy(pymt->out_paym.s_D_STATE,"TX");
    strcpy(pymt->out_paym.s_D_ZIP,"785598767");
    strcpy(pymt->out_paym.s_C_FIRST,"John");
    strcpy(pymt->out_paym.s_C_MIDDLE,"P");
    strcpy(pymt->out_paym.s_C_LAST,"Williams");
    strcpy(pymt->out_paym.s_C_STREET_1,"North Rab Road");
    strcpy(pymt->out_paym.s_C_STREET_2,"Apt 343");
    strcpy(pymt->out_paym.s_C_CITY,"La Fiera");
    strcpy(pymt->out_paym.s_C_STATE,"TX");
    strcpy(pymt->out_paym.s_C_ZIP,"785585432");
    strcpy(pymt->out_paym.s_C_PHONE,"1234567890123456");
    pymt->out_paym.s_C_SINCE_time = 0;
    strcpy(pymt->out_paym.s_C_CREDIT,"GC");
    strcpy(pymt->out_paym.s_C_DATA,"Great Ebay");
    dataSet = 0;
}

return OK;
}

extern "C" NULLDB_API int do_ords(struct ords_wrapper *ords,void
*ctx)
{
    ords->out_ords.s_transtatus = 0;

    if (dataSet == 0)
    {
        ords->out_ords.s_C_BALANCE = 100000;
        ords->out_ords.s_C_ID = 3;
        ords->out_ords.s_O_ID = 1696;
        ords->out_ords.s_O_CARRIER_ID = 9;
        ords->out_ords.s_ol_cnt = 6;
        ords->out_ords.s_O_ENTRY_D_time = 1234567890;

        strcpy(ords->out_ords.s_C_FIRST,"Homer");
        strcpy(ords->out_ords.s_C_MIDDLE,"J");
        strcpy(ords->out_ords.s_C_LAST,"Simpson");

        ords->out_ords.item[0].s_OL_AMOUNT = 30000;
        ords->out_ords.item[0].s_OL_I_ID = 23492;
        ords->out_ords.item[0].s_OL_SUPPLY_W_ID = 9;
        ords->out_ords.item[0].s_OL_QUANTITY = 5;
        ords->out_ords.item[0].s_OL_DELIVERY_D_time = 1234567890;

        ords->out_ords.item[1].s_OL_AMOUNT = 12300;
        ords->out_ords.item[1].s_OL_I_ID = 18860;
        ords->out_ords.item[1].s_OL_SUPPLY_W_ID = 9;
        ords->out_ords.item[1].s_OL_QUANTITY = 5;
        ords->out_ords.item[1].s_OL_DELIVERY_D_time = 1234567890;

        ords->out_ords.item[2].s_OL_AMOUNT = 15000;
        ords->out_ords.item[2].s_OL_I_ID = 90488;
        ords->out_ords.item[2].s_OL_SUPPLY_W_ID = 9;
        ords->out_ords.item[2].s_OL_QUANTITY = 5;
        ords->out_ords.item[2].s_OL_DELIVERY_D_time = 1234567890;

        ords->out_ords.item[3].s_OL_AMOUNT = 25000;
    }
}

```

```

ords->out_ords.item[3].s_OL_I_ID = 22741;
ords->out_ords.item[3].s_OL_SUPPLY_W_ID = 9;
ords->out_ords.item[3].s_OL_QUANTITY = 5;
ords->out_ords.item[3].s_OL_DELIVERY_D_time = 1234567890;

ords->out_ords.item[4].s_OL_AMOUNT = 20000;
ords->out_ords.item[4].s_OL_I_ID = 92952;
ords->out_ords.item[4].s_OL_SUPPLY_W_ID = 9;
ords->out_ords.item[4].s_OL_QUANTITY = 5;
ords->out_ords.item[4].s_OL_DELIVERY_D_time = 1234567890;

ords->out_ords.item[5].s_OL_AMOUNT = 2345;
ords->out_ords.item[5].s_OL_I_ID = 29956;
ords->out_ords.item[5].s_OL_SUPPLY_W_ID = 9;
ords->out_ords.item[5].s_OL_QUANTITY = 5;
ords->out_ords.item[5].s_OL_DELIVERY_D_time = 1234567890;

dataSet = 1;
}
else
{
ords->out_ords.s_C_BALANCE = 123000;
ords->out_ords.s_C_ID = 856;
ords->out_ords.s_O_ID = 418;
ords->out_ords.s_O_CARRIER_ID = 10;
ords->out_ords.s_ol_cnt = 5;
strcpy(ords->out_ords.s_C_FIRST,"Erick");
strcpy(ords->out_ords.s_C_MIDDLE,"J");
strcpy(ords->out_ords.s_C_LAST,"Forman");
ords->out_ords.s_O_ENTRY_D_time = 1234567890;

ords->out_ords.item[0].s_OL_AMOUNT = 12000;
ords->out_ords.item[0].s_OL_I_ID = 54602;
ords->out_ords.item[0].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[0].s_OL_QUANTITY = 5;
ords->out_ords.item[0].s_OL_DELIVERY_D_time = 1234567890;

ords->out_ords.item[1].s_OL_AMOUNT = 2300;
ords->out_ords.item[1].s_OL_I_ID = 18860;
ords->out_ords.item[1].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[1].s_OL_QUANTITY = 5;
ords->out_ords.item[1].s_OL_DELIVERY_D_time = 1234567890;

ords->out_ords.item[2].s_OL_AMOUNT = 56009;
ords->out_ords.item[2].s_OL_I_ID = 90488;
ords->out_ords.item[2].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[2].s_OL_QUANTITY = 5;
ords->out_ords.item[2].s_OL_DELIVERY_D_time = 1234567890;

ords->out_ords.item[3].s_OL_AMOUNT = 98000;
ords->out_ords.item[3].s_OL_I_ID = 22741;
ords->out_ords.item[3].s_OL_SUPPLY_W_ID = 10;
ords->out_ords.item[3].s_OL_QUANTITY = 5;
ords->out_ords.item[3].s_OL_DELIVERY_D_time = 1234567890;

ords->out_ords.item[4].s_OL_AMOUNT = 25000;
ords->out_ords.item[4].s_OL_I_ID = 92952;
ords->out_ords.item[4].s_OL_SUPPLY_W_ID = 10;
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_dlvly(struct dlvly_wrapper *dlvly,void
*ctx)
{
dlvly->out_dlvly.s_transtatus = 0;

```

```

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

for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
dlvly->out_dlvly.s_O_ID[districtIndex]= 2055;
}
else
{
for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
dlvly->out_dlvly.s_O_ID[districtIndex]= 2056;

dataSet = 0;
}
return OK;
}

extern "C" NULLDB_API int do_stok(struct stok_wrapper *stok,void
*ctx)
{
stok->out_stok.s_transtatus = 0;

if (dataSet == 0)
{
stok->out_stok.s_low_stock = 100;

dataSet = 1;
}
else
{
stok->out_stok.s_low_stock = 40;

dataSet = 0;
}
return OK;
}

```

nullDB/stdafx.h

```

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

#pragma once

#define WIN32_LEAN_AND_MEAN // Exclude rarely-used
stuff from Windows headers
// Windows Header Files:
#include <windows.h>
// TODO: reference additional headers your program requires here

```

nullDB/stdafx.cpp

```

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

#include "stdafx.h"

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

```

tpccIsapi/htmlPhraser.h

```

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

```

```

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

#pragma once

#include <memory.h>

// Definitions

#define NULL 0

#define COMMAND_ID 0
#define TERM_ID 1
#define W_ID 2
#define D_ID 3
#define C_ID 4
#define C_NAME 5

#define C_W_ID 6
#define C_D_ID 7
#define AMT_PAID 8

#define STK_THRESHOLD 9
#define CARRIER_NUM 10

#define ITEM_LIST_START 11
#define ITEM_LIST_FINISH 55

#define MAX_QUERY_ID 55
#define MAX_FIELD_LEN 256
#define MAX_FIELD_NUM 56

// Command Codes

#define NEW_ORDER_CODE 'n'
#define PAYMENT_CODE 'p'
#define ORDER_STATUS_CODE 'o'
#define DELIVERY_CODE 'd'
#define STOCK_CODE 's'
#define EXIT_CODE 'e'
#define MENU_CODE 'm'

#define COMMAND_LOGIN 0
#define COMMAND_NEW_ORDER 1
#define COMMAND_PAYMENT 2
#define COMMAND_ORDER_STATUS 3
#define COMMAND_DELIVERY 4
#define COMMAND_STOCK 5
#define COMMAND_EXIT 6

#define COMMAND_LOGIN_RESULTS 7
#define COMMAND_NEW_ORDER_RESULTS 8
#define COMMAND_PAYMENT_RESULTS 9
#define COMMAND_ORDER_STATUS_RESULTS 10
#define COMMAND_DELIVERY_RESULTS 11
#define COMMAND_STOCK_RESULTS 12

// Class htmlPhraser

class htmlPhraser
{
    // Constructors / Destructor
public:
    htmlPhraser(char *queryString);

```

```

~htmlPhraser() {return;}

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

    char * get_TERM_ID() {return
iQueryValues[TERM_ID];}
    char * get_W_ID() {return
iQueryValues[W_ID];}
    char * get_D_ID() {return
iQueryValues[D_ID];}
    char * get_C_ID() {return
iQueryValues[C_ID];}
    char * get_C_NAME() {return
iQueryValues[C_NAME];}
    char * get_C_W_ID() {return
iQueryValues[C_W_ID];}
    char * get_C_D_ID() {return
iQueryValues[C_D_ID];}
    char * get_AMT_PAID() {return
iQueryValues[AMT_PAID];}
    char * get_STK_THRESHOLD() {return
iQueryValues[STK_THRESHOLD];}
    char * get_CARRIER_NUM() {return
iQueryValues[CARRIER_NUM];}

    char * get_ITEM_SUPP_W(int item) {return
iQueryValues[(ITEM_LIST_START + 0) + (item * 3)];}
    char * get_ITEM_ITEM_NUM(int item) {return
iQueryValues[(ITEM_LIST_START + 1) + (item * 3)];}
    char * get_ITEM_QTY(int item) {return
iQueryValues[(ITEM_LIST_START + 2) + (item * 3)];}

// Class Functions
private:
    char convertQueryToken(char **queryString);

// Class Attributes
private:
    int iCustomerIdFlag;
    int iCarrierNumFlag;
    int iStockThresholdFlag;

    char
iQueryValues[MAX_FIELD_NUM][MAX_FIELD_LEN];
};

// tpccIsapi/resource.h

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

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

```

tpccIsapi/StdAfx.h

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

#pragma once

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

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some
CString constructors will be explicit

#define _ATL_ALL_WARNINGS

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

#ifndef _WIN32_WINNT
#define _WIN32_WINNT 0x0403
#endif

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

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

// TODO: reference additional headers your program requires here
```

tpccIsapi/tpcc.h

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

// standard includes

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

#include <db2tpcc.h>
#include <iostream>
#include <fstream>
#include <process.h>
#include <ios>

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

#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
// String Field Lengths

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

#define CREDIT_LEN            2

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

#define PHONE_LEN             16
#define DATA_LEN             200

#define ITEM_LIST             15
#define ORDER_LIST            10

// Type definitions
// Type definitions

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

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

typedef INT16b                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
// Date and time values

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

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

// Error codes
// Error codes
#define ERR_INVALID_TXN_TYPE   -1

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

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

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

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

```

```

#define ERR_C_ID_OR_CLAST_ONLY      "ERROR: Either
customer id or customer last name can be specified."

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

#define ERR_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;
        shorts_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[20];
};

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
//           appended
//           to the html page
// Comments  :
//

```

```

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

    **string='\0';
}

```

```

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

```

```

inline void appendCustData(char **string,char *text)
{
    short byteCount = 0;
    while(*text)
    {
        *(*string)++ = *text++;
        byteCount++;
        if((byteCount % 50) == 0)
        {
            *(*string)++ = '\n';
            *(*string)++ = ' ';    *(*string)++ = ' ';    *(*string)++
            = ' '; *(*string)++ = ' ';
            *(*string)++ = ' ';    *(*string)++ = ' ';    *(*string)++
            = ' '; *(*string)++ = ' ';
            *(*string)++ = ' ';    *(*string)++ = ' ';    *(*string)++
            = ' ';
        }
    }
    **string='\0';
}

```

```

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

```

```

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

```

```

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

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

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

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

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

    int daysUsed = (years * DAYS_IN_YEAR) + leaps;

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

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

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

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

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

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

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

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

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

```

```

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

//
// copyOutZip
//
// Title : Copy zip data out of class array
// Parameters : char * - buffer to copy zip string into
//
// Return Value: int - Length of copy
// Comments :
//

inline int copyOutZip(char *buffer,char *value,int len =
DEFAULT_ZIP_LEN)
{
    int index = 0;
    int bufferPos = 0;

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

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

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

    buffer[bufferPos] = NULL;
    return len;
}

//
// copyOutPhone
//
// Title : Copy phone data out of class array
// Parameters : char * - buffer to copy phone string into
//
// Return Value: int - Length of copy
// Comments :
//

inline int copyOutPhone(char *buffer,char *value,int len =
DEFAULT_PHONE_LEN)
{
    int index = 0;
    int bufferPos = 0;

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

```

```

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

buffer[bufferPos] = '\0';

return len;
}

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

    bool negativeFlag = false;

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

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

            default:
                // adjust decimal places
                decimal = decimal * 10;

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

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

```

```

        index++;
    }

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

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

    return true;
}

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

    int decimal=0;

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

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

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

    return amount;
}

```

```

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

    int             places         = 0;

    bool            negativeFlag   = false;
    bool            moneyFlag      = true;

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

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

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

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

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

```

```

                buffer[--index] =
NEGATIVE_SYMBOL;
            }
            else
            {
                // add the money indicator
                if(moneyFlag)
                {
                    moneyFlag = false;
                    buffer[--index] =
MONEY_SYMBOL;
                }
                else buffer[--index] =
NUMERIC_FILL;
            }
        }
    }

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

//return len;
}

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

    *buffer = NULL;

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

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

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

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

```

```

*buffer++ = TIME_DELIMITER;

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

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

    int           places         = 0;

    bool          negativeFlag   = false;

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

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

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

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

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

```

```

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

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

return len;
}

////////////////////////////////////
// Macros
////////////////////////////////////
using namespace std;

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

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

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

        fileName--;
    }

    return filePath;
}

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

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

    \

    \

    debugFileName(__FILE__)
    errorStream <<
    << "|" << __TIMESTAMP__ << "|" <<
    __LINE__ << "|" \
    << _getpid() << "|" <<
    GetCurrentThreadId() << "|" \
    << TEXT;

    \

    errorStream.flush();

    \

    LeaveCriticalSection(&errorMutex);

#ifdef _DEBUG

```

```

#define DEBUGMSG(TEXT)
    \

    EnterCriticalSection(&debugMutex);

    \

    \

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

    \

    debugStream.flush();

    \

    LeaveCriticalSection(&debugMutex);

#define DEBUGSTRING(TEXT,LENGTH)
    \

    debugVarString(TEXT,LENGTH)

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

#endif
#endif /* _COMMON_TPCC */

tpccIsapi/tpccIsapi.def
; tpccIsapi.def : declares the module parameters for the DLL.

LIBRARY "tpccIsapi"

EXPORTS
    HttpExtensionProc
    GetExtensionVersion
    TerminateExtension

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

#ifdef __tpccISAPI_hpp__
#define __tpccISAPI_hpp__

#include <windows.h>
#include <httpext.h>

#include <tpcc.h>

```

```

#include <htmlPhraser.h>
#include <iomanip>

#include <db2tpcc.h>
#include <comsvcs.h>

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

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

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

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

    COM_HANDLE comInterface;
};

struct DLVYQUEUEDATA
{
    int      warehouse;
    shortin_s_0_CARRIER_ID;
    struct _timeb enqueueTime;
};

////////////////////////////////////////////////////////////////
// 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\r\nW_ID: %d CARRIER_ID: %d
%s\nend-time: %d.%03d\n"

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

```

```

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

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

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

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

int appendButtons(char *htmlPage);
int appendItems(char *htmlPage, short itemCount, short 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 consutant parts
////////////////////////////////////////////////////////////////

#include "htmlPhraser.h"

////////////////////////////////////////////////////////////////
// htmlPhraser::htmlPhraser

```

```

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

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

    // this initializes the query list to NULL's. This means that
    // characters being added are overwriting null characters and
    // therefore the string will be null terminated implicitly.

    memset(iQueryValues,NULL,(MAX_FIELD_NUM *
MAX_FIELD_LEN));

    // controls
    char    queryChar    = NULL;

    int     queryIndex   = -1;
    int     valueIndex   = -1;

    // process each character of query string
    while(*queryString)
    {
        // check for special case characters
        if(queryChar)
        {
            // a percentage sign would indicate a token
            if(*queryString != '%')
            {
                // a plus sign represents a space
                if(*queryString == '+')
                {
                    queryChar = ' ';
                    *queryString++;
                }
                else queryChar = *queryString++;
            }
            else queryChar = convertQueryToken(&queryString);
        }
        else queryChar = '&';

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

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

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

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

            // finished processing for query reference
            continue;

```

```

}

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

// before accepting value

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

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

    // finishes looking 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 DLVYQUEUEDATA *dlvyQueue;         //dlvy
queue

```

```

HANDLE                *dIvyThreadHandles;        //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 dIvyLogPath[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)(dIvy_wrapper *dIvy,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        dIvyCall;

////////////////////////////////////
// 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);

    // Initialize 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(dlvy_wrapper));
maxDataSize = max(maxDataSize,sizeof(stok_wrapper));
maxDataSize += 10;

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

        return true;
    }

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

```

```

/*
*****
** Name      :    initTxnHandle
** Description :
**           Isapi thread initializes its own com
interface
**           structure.
** Parameters :
**           TXN_HANDLE**      isapi txn handle
** Returns   :
**           int - return code
** Comments  :
*****
*/
int initTxnHandle(TXN_HANDLE **txnHandle)
{
    DEBUGMSG("Inside init txn handle, getting isapiLock." <<
endl);
    EnterCriticalSection(&isapiLock);

    HRESULT hres = NULL;
    try
    {
        DEBUGMSG("Got 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 Colnitalize with txnHandle:
"<<DEBUGADDRESS(*txnHandle)<<endl);
        hres = ColnitalizeEx(NULL, COINIT_MULTITHREADED);
        if (FAILED(hres))
        {
            ERRORMSG("ColnitalizeEx() 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_I
tpcc_com, (void **)(*txnHandle)->comInterface.comHandle);
        if (FAILED(hres))
        {
            _ftime(&endTime);
            //store error code in txnHandle

```

```

            ERRORMSG("CoCreateInstance() failed,
code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<
" hres:"<<hres<< " time waiting:"<<
(((endTime.time - startTime.time)*1000)+
(endTime.millitm -
startTime.millitm))/1000.0)<<endl);

            DEBUGMSG("CoCreateInstance() failed,
code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<
" hres:"<<hres<< " time waiting:"<<
(((endTime.time - startTime.time)*1000)+
(endTime.millitm -
startTime.millitm))/1000.0)<<endl);

            return(ERR);
        };

        _ftime(&endTime);
        DEBUGMSG("CoCreateInstance successful.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 exeception in initTxnHandle,
unlocking isapi lock" <<endl);
        ERRORMSG("Unhandled exeception in initTxnHandle,
unlocking isapi lock" <<endl);
        LeaveCriticalSection(&isapiLock);
    };

    return ERR;
}

/*
*****
** Name      :    getDBInstance
** Description :
**           load db specific lib based on dbType
registry

```

```

**                               value.
** Parameters :
**
** Returns      :
**              int - return code
** Comments    :
**              This function only exists for the dlvy
threads
**              Dlvy threads hold direct connections to the
database
**              and therefore need to know what db
interface to talk to.
*****
*/
int getDBInstance()
{
    if(nullDB)
    {
        dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\nullDB.dll");
        if(dbInstance == NULL)
        {
            return ERR_NULL_DLL_NOT_LOADED;
        }
    }
    else if( (strcmp(dbType,"DB2") == 0) )
    {
        dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\tpccDB2glue.dll");
        if(dbInstance == NULL)
        {
            return ERR_DB2_DLL_NOT_LOADED;
        }
    }
    else if( (strcmp(dbType,"ORACLE") == 0) )
    {
        return ERR_ORACLE_DLL_NOT_LOADED;
    }
    else
    {
        return ERR_UNKNOWN_DB;
    }

    db_connect =
(CONNECT_PTR)GetProcAddress(dbInstance,"connect_db");
    if(db_connect == NULL)
    {
        return ERR_CONNECT_ADDRESS_NOT_FOUND;
    }
    dlvyCall =
(DLVY_FUNC_PTR)GetProcAddress(dbInstance,"do_dlvy");
    if(dlvyCall == NULL)
    {
        return ERR_DLVY_ADDRESS_NOT_FOUND;
    }

    return OK;
}

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

```

```

int initDlvy()
{
    // Initialize critical section
    InitializeCriticalSection(&dlvyQueueLock);

    //create dlvy queue
    dlvyQueue = (DLVYQUEUEUEDATA *)
calloc(dlvyQueueLen,sizeof(DLVYQUEUEUEDATA));

    dlvyThreadDone = CreateEvent(NULL,
                                TRUE, //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_SU
B_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,(BYT
E *)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;
    else
        nullDB = 0;

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

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

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

    //get the client null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYT
E *)&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,&regTy
pe,(BYTE *) &value,&regValueSize)== ERROR_SUCCESS )
            strcpy(dlvyLogPath,value);
        else
            strcpy(dlvyLogPath,DEFAULT_DLVY_LOG_PATH);

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

        //get global error log file path/name
        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,HTML_TRACE_LOG_FILE,0,&reg
Type,(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>\n"
" <BODY><FORM ACTION=\""
APP_NAME
"\" METHOD=\""GET\"">\n"
" <H3>Please Select
Transaction.</H3>\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"
//
//          "12345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890\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"
//
//          "12345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890\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, commandBl
ock, txnHandle);
        return OK;
    }
    DEBUGMSG("nord d_id:" << nord->in_nord.s_D_ID << endl);

    if((nord->in_nord.s_C_ID = atoi(commandBlock->get_C_ID())
== 0)
    {
        doNewOrderErrorPage(html, ERR_INVALID_C_ID, commandBl
ock, txnHandle);
        return OK;
    }
    DEBUGMSG("nord c_id:" << nord->in_nord.s_C_ID << endl);

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

    for (int itemIndex=0; itemIndex<NORD_ITEMS; itemIndex++)
    {
        //supply warehouse
        if( *(commandBlock->get_ITEM_SUPP_W(itemIndex))
            if ( (nord->in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID =
atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
            {
                doNewOrderErrorPage(html, ERR_INVALID_SUPPLY_W_ID, c
ommandBlock, 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,comm
andBlock,txnHandle);
                return OK;
            }
            else
                itemComplete++;
        }
        //missing previous value of item supp warehouse,
flag error
        else
        {
            doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,c
ommandBlock,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,comm
andBlock,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,comm
andBlock,txnHandle);
                    return OK;
                }
                else
                    itemComplete++;
            }
            //missing previous value of item number
            else if (itemComplete ==1)
            {
                doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,comm
andBlock,txnHandle);
                return OK;
            }
            //missing 1st value of supp warehouse
            else
            {
                doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,c
ommandBlock,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,comm
andBlock,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"
//
"123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890\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"
//
//
"123456789012345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890\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> <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"
"12345678901234567890123456789012345678901234567890123
4567890123456789012345678901234567890\r\n"
"Warehouse: ");*/
    appendText(&html,"<PRE>Warehouse: ");
    appendText(&html,itoa(txnHandle->w_id,buffer,10),7,1);
    appendText(&html,"District: <INPUT NAME=\\"
                CMD_D_ID
                "\\ SIZE=1>          Date:<BR>"
                "Customer <INPUT NAME=\\"
                CMD_C_ID
                "\\ SIZE=6> Name:          Credit:
%Disc.:<BR>"

```

```

"Order Number:      Number of Lines:
W_tax:      D_tax:<BR><BR>"
//      1      2      3      4      5
6      7      8      9\r\n"
//"1234567890123456789012345678901234567890123456789012
34567890123456789012345678901234567890\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,itoa(txnHandle->w_id,buffer,10));

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

```



```

    appendText(&html,pymt-
>out_paym.s_D_STREET_1,STREET_LEN,1);
    appendText(&html,"<BR>");

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

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

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

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

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

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

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

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

    appendText(&html,"Since: ");
    copyOutDateTIme(buffer,pymt->out_paym.s_C_SINCE_time);
    appendText(&html,buffer);

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

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

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

    appendText(&html,pymt-
>out_paym.s_C_STREET_2,STREET_LEN+21,1);
    appendText(&html,"%Disc: ");
    html+=sprintf(html,"%2.2lf",pymt-
>out_paym.s_C_DISCOUNT/100.0);

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

```

```

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

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

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

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

    appendText(&html," <BR> <BR>Amount Paid: $");
    html+=sprintf(html,"%-9.2lf",pymt-
>in_paym.s_H_AMOUNT/100.0);

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

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

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

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

    return OK;
}
/*
*****
** Name          : doPaymentErrorPage
** Description   :
**               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> 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,command
Block,txnHandle);
    return OK;
}

if( (ords->in_ords.s_C_ID = atoi(commandBlock->get_C_ID()))
== 0)
{
    if(*(commandBlock->get_C_NAME()) == NULL)
    {
        //no customer id nor customer last name specified.

doOrderStatusErrorPage(html,ERR_MISSING_C_ID_OR_CLA
ST,commandBlock,txnHandle);
        return OK;
    }
    else
        strcpy(ords->in_ords.s_C_LAST,commandBlock-
>get_C_NAME());
    }
    else
    {
        //make sure that the user only inserted just c_id
        if(*(commandBlock->get_C_NAME()) != NULL)
        {

doOrderStatusErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,
commandBlock,txnHandle);
            return OK;
        }
    }

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

    html+=appendHiddenFields(html,txnHandle);

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

    ords->out_ords.s_transtatus = -1;

    HRESULT    hres;
    try
    {
        hres = txnHandle->comInterface.comHandle-
>doOrderStatus(&txnHandle-
>comInterface.size,(UCHAR**)&txnHandle-
>comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: ords com call caused
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)<<endl);
        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,itoa(ords-
>in_ords.s_W_ID,buffer,10),6+1,1);
appendText(&html,"District: ");
appendText(&html,itoa(ords->in_ords.s_D_ID,buffer,10));
appendText(&html,"<BR>"
    "Customer: ");

//get customer id
appendText(&html,itoa(ords-
>in_ords.s_C_ID,buffer,10),6+1,1);
appendText(&html,"Name: ");
//get first, middle, and last from wrapper
appendText(&html,ords-
>out_ords.s_C_FIRST,FIRST_NAME_LEN+1,1);
appendText(&html,ords-
>out_ords.s_C_MIDDLE,INITIALS_LEN+1,1);
appendText(&html,ords-
>out_ords.s_C_LAST,LAST_NAME_LEN+5,1);

```

```

//get customer balance from wrapper
appendText(&html,"\\n\\nCust-Balance: $");
html+=sprintf(html,"%2f",ords-
>out_ords.s_C_BALANCE/100.0);

//display order number, entry date, and carrier number
appendText(&html,"<BR> <BR>"
"Order-Number ");
appendText(&html,itoa(ords-
>out_ords.s_O_ID,buffer,10),12,1);
appendText(&html,"Entry-Date: ");
copyOutDateTime(buffer,ords-
>out_ords.s_O_ENTRY_D_time);
appendText(&html,buffer,22,1);

appendText(&html,"Carrier-Number: ");
appendText(&html,itoa(ords-
>out_ords.s_O_CARRIER_ID,buffer,10));

//add item title columns
appendText(&html,"<BR>"
"Supply-W "
"Item-Id "
"Qty "
"Amount "
"Delivery-Date<BR> ");

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

//get supp w
appendText(&html,itoa(ords-
>out_ords.item[itemCount].s_OL_SUPPLY_W_ID,buffer,10),11,1);

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

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

//get item dollar amount
html+=sprintf(html,"$%-14.2f",ords-
>out_ords.item[itemCount].s_OL_AMOUNT/100.0);

//get delivery date
copyOutDate(buffer,ords-
>out_ords.item[itemCount].s_OL_DELIVERY_D_time);
appendText(&html,buffer);
appendText(&html," <BR> ");

}

//append line breaks if item count is less than 15
for (int itemCount=0;itemCount < (15-ords-
>out_ords.s_ol_cnt);itemCount++)
appendText(&html,"<BR> ");

html+=displayStatus(html,rc);

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

return OK;
}
/*
*****
** Name : doOrderStatusErrorPage

```

```

** Description :
** HTML orderStatus error page
** Parameters :
** char * html page result
** char * error message
** htmlPhraser* command block
** TXN_HANDLE* txn handle
** Returns :
** int - return code
** Comments :
**
*****
*/

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

appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order
Status</TITLE></HEAD>\\n\\n"
"<BODY><FORM ACTION=\""
APP_NAME
"\" METHOD=\"GET\">\\n\\n"
"<CENTER><H3>Please Fill In Order
Status Form.</H3></CENTER> <BR>\\n\\n"
"Submit Transaction <INPUT
TYPE=\"submit\" NAME=\""
CMD_TXN_ID
"\" VALUE=\""
CMD_ORDS
"\">"
"<BR> ");
html+=appendHiddenFields(html,txnHandle);

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

char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));

appendText(&html," District: <INPUT NAME=\""
CMD_D_ID
"\" SIZE=1>\\n<BR>"
"Customer: "
"<INPUT NAME=\""
CMD_C_ID
"\" SIZE=5>"
" "
"Name: "
"<INPUT NAME=\""
CMD_C_NAME
"\" SIZE=20><BR>"
"Cust-Balance: <BR>"
"Order-Number: Entry-Date:
Carrier-Number<BR>"
"Supply-W Item-Num Qty
Amount Delivery <BR>");

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

return OK;
}
/*
*****
** Name : doDeliveryForm
** Description :
** HTML payment page entry point
** Parameters :
** htmlPhraser* command block
** TXN_HANDLE* txn handle struct

```

```

** Returns      :
**              int - return code

** Comments    :
**
*****
*/
int doDeliveryForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=txnHandle->htmlPage;

    appendText(&html, "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD>\r\n"
               "<BODY><FORM ACTION=\""
               APP_NAME
               "\" METHOD=\"GET\">\r\n"

               "<CENTER><H3>Delivery.</H3></CENTER>\r\n"
               "Submit Transaction <INPUT
TYPE=\"submit\" NAME=\""
               CMD_TXN_ID
               "\" VALUE=\""
               CMD_DLVY
               "\">");
    html+=appendHiddenFields(html, txnHandle);

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

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

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

    return OK;
}

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

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

    //declare delivery structure
    struct dlvy_wrapper    dlvy;

    //set warehouse login id from command blk
    dlvy.in_dlvly.s_W_ID = txnHandle->w_id;

    //set the carrier id from command blk

```

```

    if( (dlvy.in_dlvly.s_O_CARRIER_ID = atoi(commandBlock-
>get_CARRIER_NUM())) == 0)
    {

        doDeliveryErrorPage(html, ERR_INVALID_CARRIER, command
Block, txnHandle);
        return OK;
    }

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

    html+=appendButtons(html);

    html+=appendHiddenFields(html, txnHandle);

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

    int rc =
    queueDlvyTxn(dlvy.in_dlvly.s_W_ID, dlvy.in_dlvly.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_dlvly.s_W_ID, buffer, 10));
    appendText(&html, "<BR> <BR>Carrier Number: ");

    //get carrier_id from wrapper
    appendText(&html, itoa(dlvy.in_dlvly.s_O_CARRIER_ID, buffer, 1
0));
    appendText(&html, "<BR> <BR>Execution Status: Delivery has
been queued </PRE></BODY></HTML>");

    return OK;
}

/*
*****
** Name          : doDeliveryErrorPage
** Description   :
**              HTML payment error page entry point
** Parameters    :
**              char *      html result page
**              char *      error message
**              htmlPhraser command block
**              TXN_HANDLE*  txn handle
**
** Returns      :
**              int - return code
** Comments     :
**
*****
*/
int doDeliveryErrorPage(char *htmlPage, char
*message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{

```

```

char *html=htmlPage;

appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD>\r\n"
" <BODY><FORM ACTION=\"
APP_NAME
\" METHOD=\"GET\">\r\n"

"<CENTER><H3>Delivery.</H3></CENTER>\r\n"
"Submit Transaction <INPUT
TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\" VALUE=\"\"
CMD_DLVS
\">");
html+=appendHiddenFields(html,txnHandle);

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

char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));

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

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

return OK;
}

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

int doStockForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
char *html=txnHandle->htmlPage;
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\r\n"
" <BODY><FORM ACTION=\"
APP_NAME
\" METHOD=\"GET\">\r\n"
"<CENTER><H3>Please Fill In Stock
Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction <INPUT
TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\" VALUE=\"\"
CMD_STOK
\">");
html+=appendHiddenFields(html,txnHandle);

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

```

```

char buffer[15];
appendText(&html,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,comman
dBlock,txnHandle);
return OK;
}

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

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

html+=appendButtons(html);

html+=appendHiddenFields(html,txnHandle);

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

```

```

    stok->out_stok.s_transtatus = -1;

    DEBUGMSG("Calling com entry api for stock call,
w_id:"<<stok->in_stok.s_W_ID<<" d_id:"<<stok->in_stok.s_D_ID<<
    " threshold:"<<stok->in_stok.s_threshold<<endl);

    HRESULT hres;
    try
    {
        hres = txnHandle->comInterface.comHandle-
>doStockLevel(&txnHandle-
>comInterface.size,(UCHAR**)&txnHandle-
>comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: Com Stock call caused
exeception to occur.</PRE></BODY></HTML>");
        ERRORMSG("ERROR : Com Stock call caused
exeception to occur."<<endl);
        return OK;
    }

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

    hres = txnHandle->comInterface.comHandle-
>doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: stok com doSetComplete
failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : stok com doSetComplete failed,
rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    stok = (stok_wrapper *)txnHandle->comInterface.txnBuffer;
    int rc = stok->out_stok.s_transtatus;
    if(rc != OK)
    {
        html+=displayStatus(html,rc);
        appendText(&html,"</PRE></BODY></HTML>");
        ERRORMSG("ERROR stok txn failed"<<endl
        <<"stok->in_stok.s_D_ID:"<<stok-
>in_stok.s_D_ID<<endl
        <<"stok->in_stok.s_threshold:"<<stok-
>in_stok.s_threshold<<endl
        <<"stok->in_stok.s_W_ID:"<<stok-
>in_stok.s_W_ID<<endl
        <<"stok->out_stok.deadlocks:"<<stok-
>out_stok.deadlocks<<endl
        <<"stok->out_stok.s_low_stock:"<<stok-
>out_stok.s_low_stock<<endl
        <<"stok->out_stok.s_transtatus:"<<stok-
>out_stok.s_transtatus<<endl);
        return OK;
    }

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

    //get w_id from wrapper

```

```

char buffer[10];
appendText(&html,itoa(stok->in_stok.s_W_ID,buffer,10),6+1,1);

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

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

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

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

return OK;
}

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

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

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

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

    char buffer[15];
    appendText(&html,itoa(txnHandle->w_id,buffer,10));
    appendSpaces(&html,2);
    appendText(&html,"District: ");
    appendText(&html,commandBlock->get_D_ID());
    appendText(&html," <BR> <BR>"
        "Stock Level Threshold: "
        "<INPUT NAME=\\"
        CMD_STK_THRESHOLD
        "\" SIZE=1> <BR> <BR>"
        "Low Stock: <BR>");

```

```

appendText(&html,message);

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

return OK;
}

/*
*****
** Name      : doExit
** Description :
**           HTML exit page entry point
** Parameters :
**           htmlPhraser* command block
**           TXN_HANDLE*  txn handle struct
** Returns   :
**           int - return code
** Comments  :
**
*****
*/
int doExit(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    return (doLoginForm(commandBlock,txnHandle));
}

/*
*****
** Name      : displayStatus
** Description :
**           appends status string to the html page
** Parameters :
**           char*      html page
**           int       rc
** Returns   :
**           amount of characters the function appened
**           to the html page
** Comments  :
**
*****
*/
int displayStatus(char *htmlPage,int rc)
{
    char *html = htmlPage;

    appendText(&html,"");

    switch (rc)
    {
    case OK:
        appendText(&html,"Execution Status: Transaction
Committed",50,1);
        break;
    case INVALID_ITEM:
        appendText(&html,"Execution Status: Item number is not
valid",50,1);
        break;
    case INVALID_STATUS:
        appendText(&html,"Execution Status: ERROR: Rollback
INVALID_STATUS",50,1);
        break;
    case INVALID_COM_STATUS:
        appendText(&html,"Execution Status: ERROR: Rollback
COM FAILURE",50,1);
        break;
    case ERR_DLVS_QUEUE_FULL:
        appendText(&html,"Execution Status: ERROR: Rollback
DLVS QUEUE FULL",50,1);

```

```

        break;
    default:
        appendText(&html,"Execution Status: ERROR:
Rollback",50,1);
    };

    appendText(&html," ");

    return (int)(html - htmlPage);
}
/*
*****
** Name      : appendButtons
** Description :
**           append hidden field to recognize user after
login
** Parameters :
**           *htmlPage      html result page
**           *TXN_HANDLE    txn handle
** Returns   :
**           int            amount of characters the
function appened
**           to the html 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 appened
**           to the html page
** Comments  :
**
*****
*/
int appendButtons(char *htmlPage)
{
    char *html = htmlPage;

    appendText(&html,"<INPUT TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\" VALUE=\"\"
CMD_NORD
\">\r\n\"
\"<INPUT TYPE=\"submit\" NAME=\"\"
CMD_TXN_ID
\" VALUE=\"\"
CMD_PYMT

```

```

        "\>\r\n"
        "<INPUT TYPE=\submit\ NAME=\\""
        CMD_TXN_ID
        "\ VALUE=\\""
        CMD_ORDS
        "\>\r\n"
        "<INPUT TYPE=\submit\ NAME=\\""
        CMD_TXN_ID
        "\ VALUE=\\""
        CMD_DLVY
        "\>\r\n"
        "<INPUT TYPE=\submit\ NAME=\\""
        CMD_TXN_ID
        "\ VALUE=\\""
        CMD_STOK
        "\>\r\n"
        "<INPUT TYPE=\submit\ NAME=\\""
        CMD_TXN_ID
        "\ VALUE=\\""
        CMD_EXIT
        "\>\r\n <BR>");

    return (int)(html - htmlPage);
}

/*
*****
** Name      : appendItems
** Description :
**           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 appened
**           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,dlvyThreadI
D);
    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_dlvys_0_CARRIER_ID =
            dlvyQueueData.in_s_0_CARRIER_ID;
            dlvyTxn.in_dlvys_W_ID =
            dlvyQueueData.warehouse;
            dlvyTxn.out_dlvys_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_dlvys_transtatus;

            DEBUGMSG("dlvy txn response time:"<<
            (((endProcessTime.time -
            endQueueTime.time)*1000)+
            (endProcessTime.millitm -
            endQueueTime.millitm))/1000.0)<<
            " w_id:"<<dlvyTxn.in_dlvys_W_ID<<"
            carrier:" <<dlvyTxn.in_dlvys_0_CARRIER_ID<<
            "
            deadLocks:"<<dlvyTxn.out_dlvys_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_DLVE_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_DLVE_QUEUE_FULL;
        }
    }
}

```

```

LeaveCriticalSection(&dIvYQueueLock);

//release semaphore to wake thread that there is work
ReleaseSemaphore(dIvYThreadSemaphore,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()](&commandBl
ock,txnHandle);
    DEBUGMSG("Return from html page
function:"<<commandBlock.getCommandId()<<endl);

return;
}

/*
*****
** Name          : getTerminal
** Description   :
**               retrieves terminal information based on
terminal id
** Parameters    :
**               int          terminal id
**               TERM_HANDLE* txn handle
** Returns       :
**               int - return code
** Comments      :
**
*****
*/

int getTerminal(int terminal,TXN_HANDLE *txnHandle)
{
    //check to see if terminal id is out of range
    if(terminal >= numUsers)
    {
        //terminal id not valid.
        sprintf(txnHandle->htmlPage,"ERROR: Client does not
support more than %d users, terminal id:%d",numUsers,terminal);
        ERRORMSG("ERROR : Client does not support more
than "<<numUsers<<" users, terminal id:"<<terminal<<endl);
        return ERR;
    }

    //check if terminal id is points to a not in use terminal
    if(!(termArray+terminal)->terminalInUse)
    {
        sprintf(txnHandle->htmlPage,"ERROR: Terminal id given
points to a not in use terminal.");
        ERRORMSG("ERROR : Terminal id given points to a not
in use terminal."<<endl);
        return ERR;
    }

    DEBUGMSG("Storing terminal warehouse, district , and initial
term id for user:"<<terminal<<endl);

    //assign terminal values to txn_handle
    txnHandle->d_id = termArray[terminal].d_id;
    txnHandle->w_id = termArray[terminal].w_id;
    txnHandle->term_id = termArray[terminal].terminalID;

    DEBUGMSG("Users terminal:"<<terminal<< ", stored
warehouse:"<<txnHandle->w_id<<
" district:"<<txnHandle->d_id<< " terminalID
stored:"<<txnHandle->term_id<<endl);

return OK;
}

/*
*****
** Name          : assignTerminal
** Description   :
**               assigns terminal index to user
** Parameters    :
**               TERM_HANDLE* txn handle
** Returns       :
**               int - return code

```

```

** Comments      :
**
*****
*/
int assignTerminal(TXN_HANDLE *txnHandle)
{
    EnterCriticalSection(&termLock);

    //check if terminal array is full.
    if(termNextFree == numUsers)
    {
        LeaveCriticalSection(&termLock);
        return ERR;
    }

    DEBUGMSG("Storing user warehouse:"<<txnHandle->w_id<<"
district:"<< txnHandle->d_id<<
        " in terminal slot:"<<termNextFree<<endl);

    //store users w_id and d_id
    termArray[termNextFree].d_id = txnHandle->d_id;
    termArray[termNextFree].w_id = txnHandle->w_id;

    //set terminal slot to be in use
    termArray[termNextFree].terminalInUse = true;
    termArray[termNextFree].terminalID = termNextFree;
    //in txn handle, set the terminal id
    txnHandle->term_id = termNextFree;

    //increment to next free terminal.
    termNextFree++;

    DEBUGMSG("User warehouse:"<<txnHandle->w_id<<"
district:"<< txnHandle->d_id <<
        " stored in terminal slot:"<<txnHandle->term_id<<" next
terminal free:"<<termNextFree<<endl);

    LeaveCriticalSection(&termLock);

    return OK;
}

```

A.2 Client Transaction Code

Makefile.config

```

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

#
# Makefile.config - AIX 64-bit
#

# Make Configuration
MAKE=make

```

```

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUGIT" "-g -
DDEBUGIT" or left blank
CC=xlc
CFLAGS_OS=-qflag=i:i -qlanglvl=ansi -qcpluscmt -DSQLUNIX -
DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=xlc
LD_STORP=xlc
LD_FLAGS_EXEC=-q64
LD_FLAGS_SHLIB=-qmkschrobj
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) -bE:$@.exp -lc -b64
LD_FLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2
LD_FLAGS_OUT=-o

# Library Configuration
AR=ar
AR_FLAGS=-r -v -X64
AR_FLAGS_LIB=
AR_FLAGS_OUT=

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path Separators
OBJEXT=.o
LIBEXT=.a
SHLIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;

```

tpccenv.sh

```

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

#
# tpccenv.sh - UNIX Environment Setup
#

# The Kit Version
export TPCC_VERSION=CK040318

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

# The OS being used (i.e. "UNIX", "WINDOWS")
export PLATFORM=UNIX

# The type of make command and slash used by the OS.
# (i.e. UNIX - "/", WINDOWS - "\").

```

```

# These are referenced all over the kit.
export SLASH="/";
export MAKE=make

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

export DB2VERSION=v8

# The schema name is typically the SQL authorization ID (or
username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1;      # set to the number of nodes you
have. Set to 1 for EE.

# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata

# TPCC Debug Configuration
# This is the path where all error and debug logs are placed.
# To get debugging from within the stored procedures, you must
# set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
export TPCC_DEBUGDIR=/tmp

# Specifies where stored procedures should be placed and if they
should
# be fenced.
export TPCC_SPDIR=${TPCC_SQLLIB}/function
export TPCC_FENCED=NO

include/db2tpcc.h

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

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

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

```

```

#include "lval.h"

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

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

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

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

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/* ***** */
/* NURand Constants */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause
2.1.6. */
/* Analysis indicates that a C_LAST delta of 85 is optimal.
*/
/* ***** */
#define C_C_LAST_RUN      88
#define C_C_LAST_LOAD    173
#define C_C_ID            319
#define C_OL_I_ID        3849
#define A_C_LAST          255
#define A_C_ID            1023
#define A_OL_I_ID        8191

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

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

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    int32_t s_C_ID;
    int32_t s_W_ID;
    int16_t s_D_ID;

```

```

int16_t s_O_OL_CNT;      /* init by SUT */
int16_t s_all_local;
int16_t duplicate_items;
};

struct out_neword_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
struct items_struct {
int32_t s_I_PRICE;
int32_t s_OL_AMOUNT;
int16_t s_S_QUANTITY;
int16_t pad2;
char s_I_NAME[25];
char s_brand_generic;
} item[15];
int64_t s_O_ENTRY_D_time;
int32_t s_W_TAX;
int32_t s_D_TAX;
int32_t s_C_DISCOUNT;
int32_t s_total_amount;
int32_t s_O_ID;
int16_t s_O_OL_CNT;
int16_t s_transtatus;
int16_t deadlocks;
char s_C_LAST[17];
char s_C_CREDIT[3];
};

struct in_payment_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_H_DATE_time; /* init by SUT */
int64_t s_H_AMOUNT;
int32_t s_W_ID;
int32_t s_C_W_ID;
int32_t s_C_ID;
int16_t s_C_D_ID;
int16_t s_D_ID;
char s_C_LAST[17];
};

struct out_payment_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_H_DATE_time;
int64_t s_C_SINCE_time;
int64_t s_C_CREDIT_LIM;
int64_t s_C_BALANCE;
int32_t s_C_DISCOUNT;
int32_t s_C_ID;
int16_t s_transtatus;
int16_t deadlocks;
char s_W_STREET_1[21];
char s_W_STREET_2[21];
char s_W_CITY[21];
char s_W_STATE[3];
char s_W_ZIP[10];
char s_D_STREET_1[21];
char s_D_STREET_2[21];
char s_D_CITY[21];
char s_D_STATE[3];
char s_D_ZIP[10];
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
char s_C_STREET_1[21];
char s_C_STREET_2[21];
char s_C_CITY[21];
char s_C_STATE[3];
char s_C_ZIP[10];
char s_C_PHONE[17];
};

```

```

char s_C_CREDIT[3];
char s_C_DATA[201];
};

struct in_ordstat_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_C_ID;
int32_t s_W_ID;
int16_t s_D_ID;
int16_t pad1[3];
char s_C_LAST[17];
};

struct out_ordstat_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_C_BALANCE;
int64_t s_O_ENTRY_D_time;
int32_t s_C_ID;

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

struct in_delivery_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int64_t s_O_DELIVERY_D_time; /* init by SUT */
int32_t s_W_ID;
int16_t s_O_CARRIER_ID;
};

struct out_delivery_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_O_ID[10];
int16_t s_transtatus;
int16_t deadlocks;
};

struct in_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_threshold;
int32_t s_W_ID;
int16_t s_D_ID;
};

struct out_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_low_stock;
int16_t s_transtatus;
int16_t deadlocks;
};

```

```

/* ***** */
/* Transaction Prototypes */
/* ***** */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct
out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct
out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct
out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct
out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct
out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/* ***** */
/* DB2 Connect/Disconnect & Thread Context Wrappers */
/* ***** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);

#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H

include/lval.h

#ifdef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 64000
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

include/tpccapp.h

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

```

```

** disclosure restricted by GSA ADP Schedule Contract with IBM
Corp.
***** */
/*
* tpccapp.h - Application Macros
*/

#ifdef __TPCCAPP_H
#define __TPCCAPP_H

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

#define daricall

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

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

/* *****
*
* FUNCTION: SwapEndian
* PURPOSE: Swap the byte order of a structure
* EXAMPLE: int l=0x12345678; SWAP_BYTE(l); l => 0x78563412;
* IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR
op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Add + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78

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

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

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

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

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

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

```

```

}

#endif // __TPCCAPP_H

include/tpccdbg.h

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

/*
 * tpccdbg.h - Debugging Macros
 */

#ifdef __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 tpcctx.bnd $(BND_OPTS)

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

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

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

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

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

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

```

Src.Common/tpccctx.sqc

```

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

/*
* tpcctx.sqc - TPCC context code

```

```

*/

#include <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.h"

int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username, char
*in_password);
int disconnect_from_TM(void);

int 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;

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", __FILE__, __LINE__,
&sqlca);

return ConnectSQLCODE;
}

return 0;
}

int disconnect_from_TM(void)
{
SQL_STRUCTURE sqlca sqlca;
int DisconnectSQLCODE = 0;

EXEC SQL CONNECT RESET;

DisconnectSQLCODE = SQLCODE;
if (DisconnectSQLCODE != 0) {
sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__, __LINE__,
&sqlca);
}

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

```


Src.Common/tpccdbg.c

```
/******  
*****  
** Licensed Materials - Property of IBM  
**  
** Governed under the terms of the International  
** License Agreement for Non-Warranted Sample Code.  
**  
** (C) COPYRIGHT International Business Machines Corp. 1996 -  
2004  
** All Rights Reserved.  
**  
** US Government Users Restricted Rights - Use, duplication or  
** disclosure restricted by GSA ADP Schedule Contract with IBM  
Corp.  
*****/  
  
/*  
 * tcdbg.c - Debugging Routines  
*/  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <ctype.h>  
#include <time.h>  
  
#include "sqlca.h"  
#include "sql.h"  
#include "db2tpcc.h"  
#include "tpccdbg.h"  
  
#define DEBUG_FILENAME_SZ 128  
#define DEBUG_PATH_SIZE 128  
  
void del_print();  
void new_print();  
void ord_print();  
void pay_print();  
void stk_print();  
  
void current_tmstamp(char *buf);  
  
static int debugInit = 0;  
static char debugPath[DEBUG_PATH_SIZE] = "";  
  
/*-----*/  
/* InitializeDebug */  
/*-----*/  
__inline void InitializeDebug(void) {  
    if (debugInit == 0) {  
        char *p = getenv("TPCC_DEBUGDIR");  
        if (p) {  
            strncpy(debugPath, p, DEBUG_PATH_SIZE);  
        } else {  
            strcpy(debugPath, "/tmp");  
        }  
        strcat(debugPath, "/");  
    }  
    debugInit = 1;  
}  
  
/*-----*/  
/* sqlerror */  
/*-----*/  
void sqlerror(int tranType, char *msg, char *file, int line,  
SQL_STRUCTURE sqlca *psqlca)  
{  
    FILE *err_fp = NULL;  
    char err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];  
    char tranName[16];
```

```
int j,k;  
char timeStamp[27];  
char errStr[512] = "";  
  
InitializeDebug();  
strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);  
current_tmstamp(&timeStamp[0]);  
timeStamp[19] = (char)NULL;  
  
switch(tranType)  
{  
    case NEWORD_SQL:  
        // sprintf(err_fn, "%d.err.out", getpid());  
        strcat(err_fn, "new.err.out");  
        strcpy(tranName, "NEW_ORDER");  
        break;  
  
    case DELIVERY_SQL:  
        // sprintf(err_fn, "%d.err.out", getpid());  
        strcat(err_fn, "del.err.out");  
        strcpy(tranName, "DELIVERY");  
        break;  
  
    case PAYMENT_SQL:  
        // sprintf(err_fn, "%d.err.out", getpid());  
        strcat(err_fn, "pay.err.out");  
        strcpy(tranName, "PAYMENT");  
        break;  
  
    case ORDSTAT_SQL:  
        // sprintf(err_fn, "%d.err.out", getpid());  
        strcat(err_fn, "ord.err.out");  
        strcpy(tranName, "ORDER_STAT");  
        break;  
  
    case STOCKLEV_SQL:  
        //sprintf(err_fn, "%d.err.out", getpid());  
        strcat(err_fn, "stk.err.out");  
        strcpy(tranName, "STOCK_LVL");  
        break;  
  
    case 0:  
        strcat(err_fn, "cli.err.out");  
        strcpy(tranName, "CLIENT");  
        break;  
  
    default:  
        return;  
}  
  
/* Generate Formatted Error Message */  
sqlaintp(errStr, 512, 78, psqlca);  
  
err_fp = fopen(err_fn, "a+");  
  
fprintf(err_fp, "-----\n");  
fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);  
fprintf(err_fp, "FILE %s (%u)\n", file, line);  
fprintf(err_fp, "SQLCODE %d", psqlca->sqlcode);  
fprintf(err_fp, "PID %d", getpid());  
fprintf(err_fp, "TIME %s\n", timeStamp);  
fprintf(err_fp, "-----\n");  
fprintf(err_fp, "%s", errStr);  
fprintf(err_fp, "-----\n");  
  
if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')  
{  
    fprintf(err_fp, "slerrmc: ");  
  
    for(j = 0; j < 5; j++)  
    {
```

```

for(k = 0; k < 16; k++) {
    int pos = j * 16 + k;
    if (pos < 70) fprintf(err_fp, "%02x ", psqlca->sqlerrmc[pos]);
    else fprintf(err_fp, " ");
}
fprintf(err_fp, " |");
for(k = 0; k < 16; k++) {
    int pos = j * 16 + k;
    char c = ' ';
    if (pos < 70) {
        c = psqlca->sqlerrmc[pos];
        if (!isprint(c)) c = ' ';
    }
    fprintf(err_fp, "%c", c);
}
fprintf(err_fp, "\n");
if (j < 4) fprintf(err_fp, " ");
}

fprintf(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_SIZE];

    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 = %ld (%IX)\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, "ts_O_ID[%d] = %d\n",
j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\n");
fclose(debug_fp);
}

/*-----*/
/* new_debug */
/*-----*/
void new_debug (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SIZE];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
    strcat(debug_fn, "new.debug.out");
    new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*-----*/
/* new_print */
/*-----*/
void new_print (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j, items;

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

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

```

```

return;
}

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

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

fprintf(debug_fp, "in_neword_struct {\n");

fprintf(debug_fp, "ts_C_ID      = %d (%X)\n",
in_neword->s_C_ID, in_neword->s_C_ID);
fprintf(debug_fp, "ts_W_ID      = %d (%X)\n",
in_neword->s_W_ID, in_neword->s_W_ID);
fprintf(debug_fp, "ts_D_ID      = %d (%X)\n",
in_neword->s_D_ID, in_neword->s_D_ID);
fprintf(debug_fp, "ts_O_OL_CNT   = %d (%X)\n",
in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
fprintf(debug_fp, "ts_all_local  = %d (%X)\n",
in_neword->s_all_local, in_neword->s_all_local);
fprintf(debug_fp, "ts_O_ENTRY_D  = %lld (%lX)\n",
in_neword->s_O_ENTRY_D_time, in_neword-
>s_O_ENTRY_D_time);
// fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
// in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
// in_neword->duplicate_items, in_neword->duplicate_items);

fprintf(debug_fp, "titems {\n");
items = in_neword->s_O_OL_CNT;
for (j=0; j<items; j++) {
if(j != 0)
fprintf(debug_fp, "\n");
fprintf(debug_fp, "ts_OL_I_ID[%d] = %d (%X)\n",
j, in_neword->in_item[j].s_OL_I_ID, in_neword-
>in_item[j].s_OL_I_ID);
fprintf(debug_fp, "ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
j, in_neword->in_item[j].s_OL_SUPPLY_W_ID, in_neword-
>in_item[j].s_OL_SUPPLY_W_ID);
fprintf(debug_fp, "ts_OL_QUANTITY[%d] = %d (%X)\n",
j, in_neword->in_item[j].s_OL_QUANTITY, in_neword-
>in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "\t}\n\n");

fprintf(debug_fp, "out_neword_struct {\n");
fprintf(debug_fp, "ts_C_LAST   = %s\n",
neword_ptr->s_C_LAST);
fprintf(debug_fp, "ts_C_CREDIT = %s\n",
neword_ptr->s_C_CREDIT);
fprintf(debug_fp, "ts_W_TAX    = %d\n",
neword_ptr->s_W_TAX);
fprintf(debug_fp, "ts_D_TAX    = %d\n",
neword_ptr->s_D_TAX);
fprintf(debug_fp, "ts_C_DISCOUNT = %d\n",
neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "ts_O_ID     = %d (%X)\n",
neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp, "ts_O_OL_CNT = %d (%X)\n",
neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%lX)\n",
neword_ptr->s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_total_amount = %d\n",
neword_ptr->s_total_amount);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
neword_ptr->deadlocks, neword_ptr->deadlocks);

```

```

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

fprintf(debug_fp, "titems {\n");
items = neword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
if(j != 0)
fprintf(debug_fp, "\n");
fprintf(debug_fp, "ts_I_NAME[%d] = %s\n",
j, neword_ptr->item[j].s_I_NAME);
fprintf(debug_fp, "ts_I_PRICE[%d] = %d\n",
j, neword_ptr->item[j].s_I_PRICE);
fprintf(debug_fp, "ts_OL_AMOUNT[%d] = %d\n",
j, neword_ptr->item[j].s_OL_AMOUNT);
fprintf(debug_fp, "ts_S_QUANTITY[%d] = %d (%X)\n",
j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr-
>item[j].s_S_QUANTITY);
fprintf(debug_fp, "ts_brand_generic[%d] = %c\n",
j, neword_ptr->item[j].s_brand_generic);
}
fprintf(debug_fp, "\t}\n\n");
fclose(debug_fp);
}

/*-----*/
/* ord_debug */
/*-----*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
struct in_ordstat_struct *in_ordstat,
char *msg)
{
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\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\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();
strcpy(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\n");

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

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

Src.Common/tpccmisc.c

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

/*
* tpccmisc.c - Miscellaneous routines
*/

#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);

```

```

double current_time(void);

/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
    /* use time() to get seconds */
    return(time(NULL));
}

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

```

Src.Srv/Makefile

```

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

#
# Makefile - Makefile for Src.Srv
#

include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

BND_OPTS = GRANT PUBLIC \
    MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
    EXPLAIN ALL \
    MESSAGES $.prep.msg

INCLUDE = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include

CFLAGS = $(CFLAGS_OS) $(INCLUDE) $(CFLAGS_DEBUG) \
    -D$(DB2EDITION) -D$(DB2VERSION) \
    -DSQLA_NOLINES -DLINT_ARGS

LDFLAGS = $(LDFLAGS_STORP) $(LDFLAGS_LIB)

#
#####
#####

```

```

# File Collections
#
#####
#####

STORED_PROCS = new ord del

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

EXE = news ords dels

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

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

clean: connect uncatalog unexplain disconnect
    - $(ERASE) $(TPCC_SPDIR)$(SLASH)news
    - $(ERASE) $(TPCC_SPDIR)$(SLASH)ords
    - $(ERASE) $(TPCC_SPDIR)$(SLASH)dels
    - $(ERASE) *.bnd *.msg *.out *$(OBJEXT) $(EXE)
tpcc_all_sql.c
    - $(ERASE) TPCC_ALL.*.plan

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

catalog: uncatalog
    - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl
$(STORED_PROCS)
    - db2 -tvf cat-proc.ddl +o -z cat-proc.out
    - db2 -td$$ -vf cat-func.ddl +o -z cat-func.out

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

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

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

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

plan:
    - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s
$(TPCC_SCHEMA) -w -1 -n TPCC_ALL -g -# 0 -o
TPCC_ALL.exfmt.plan

```

```
- (export DB2EXPLN_BUFFER=3000000; db2expln -d
$(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p TPCC_ALL -s 0 -g -o
TPCC_ALL.expln.plan )
```

```
rebind: connect catalog
db2 bind tpcc_all_sql.bnd $(BND_OPTS) QUERYOPT 7
```

```
#
#####
#####
# Install Targets
#
#####
#####
```

```
install: $(EXE)
- mkdir $(TPCC_SPDIR)
$(COPY) ords $(TPCC_SPDIR)
$(COPY) news $(TPCC_SPDIR)
$(COPY) dels $(TPCC_SPDIR)
```

```
#
#####
#####
# Build Rules
#
#####
#####
```

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

```
# d230437mte: QUERYOPT 7 required for UNION ALL
# Only stock needs CS , and that can be specified on the SELECT
statement
```

```
tpcc_all_sql.c:
@echo "Prepping $*.sqc"
-db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
@echo "Binding $*.bnd"
db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7
```

```
# Stored procedures are built in a special way
```

```
tpcc_all_sql$(OBJEXT):
$(CC) -c tpcc_all_sql.c $(CFLAGS) -D$(TPCC_SPTYPE)
$(CFLAGS_OUT)$@
```

```
$(EXE): $(UTIL_OBJ) tpcc_all_sql.o
$(LD_STORP) $(LD_FLAGS) $(UTIL_OBJ) tpcc_all_sql.o
$(LD_FLAGS_OUT)$@
```

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

```
# Executables (Stored Procedures)
$(EXE): $(UTIL_OBJ) tpcc_all_sql.o
```

```
# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c
```

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

Src.Srv/cat-func.ddl

-- Licensed Materials - Property of IBM

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

-- cat-func.ddl - Create table functions
--

```
--
-- DELIVERY
--
```

```
CREATE FUNCTION DEL( W_ID INTEGER
, D_ID SMALLINT
, CARRIER_ID SMALLINT
, DELIVERY_D BIGINT
)
)
```

```
RETURNS TABLE ( O_ID INTEGER )
```

```
SPECIFIC DELIVERY
```

```
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL
```

```
VAR: BEGIN ATOMIC
```

```
DECLARE O_ID INTEGER ;
DECLARE C_ID INTEGER ;
DECLARE AMOUNT INTEGER ;
```

```
/* Delete the order from new order table */
```

```
SET VAR.O_ID = ( SELECT NO_O_ID
```

```
FROM OLD TABLE ( DELETE
```

```
FROM ( SELECT NO_O_ID
```

```
FROM NEW_ORDER
```

```
WHERE NO_W_ID = DEL.W_ID
AND NO_D_ID = DEL.D_ID
```

```
ORDER BY NO_O_ID ASC
```

```
FETCH FIRST 1 ROW ONLY
) AS NEW_ORDER
) AS D
```

```
;
```

```
/* Update the order as delivered and retrieve the customer id */
```

```
SET VAR.C_ID = ( SELECT O_C_ID
```

```
FROM OLD TABLE ( UPDATE ORDERS
```

```
SET O_CARRIER_ID =
```

```
DEL.CARRIER_ID
```

```
WHERE O_W_ID = DEL.W_ID
AND O_D_ID = DEL.D_ID
AND O_ID = VAR.O_ID
```

```

        ) AS U
    )
;
SET VAR.AMOUNT = ( SELECT SUM( OL_AMOUNT )
                   FROM OLD TABLE ( UPDATE ORDER_LINE
                                     SET OL_DELIVERY_D =
DEL.DELIVERY_D
                                     WHERE OL_W_ID = DEL.W_ID
                                       AND OL_D_ID = DEL.D_ID
                                       AND OL_O_ID = VAR.O_ID
                                     ) AS U
);
/* Charge the customer */
UPDATE CUSTOMER
  SET C_BALANCE = C_BALANCE + VAR.AMOUNT
    , C_DELIVERY_CNT = C_DELIVERY_CNT + SMALLINT( 1 )
  WHERE C_W_ID = DEL.W_ID
    AND C_D_ID = DEL.D_ID
    AND C_ID = VAR.C_ID
;
/* Return the order id to the caller (or NULL) */
RETURN VALUES VAR.O_ID ;
END
$
--
-- ORDER STATUS
--
CREATE FUNCTION ORD_C_LAST( W_ID INTEGER
                          , D_ID SMALLINT
                          , C_LAST VARCHAR(16)
                          )
  RETURNS TABLE( O_ID INTEGER
                , O_CARRIER_ID SMALLINT
                , O_ENTRY_D BIGINT
                , C_BALANCE BIGINT
                , C_FIRST VARCHAR(16)
                , C_MIDDLE CHAR(2)
                , C_ID INTEGER
                )
SPECIFIC ORD_C_LAST
READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC
LANGUAGE SQL
VAR: BEGIN ATOMIC
  DECLARE C_BALANCE BIGINT ;
  DECLARE C_FIRST VARCHAR(16) ;
  DECLARE C_MIDDLE CHAR(2) ;
  DECLARE C_ID INTEGER ;
  DECLARE O_ID INTEGER ;
  DECLARE O_CARRIER_ID SMALLINT ;
  DECLARE O_ENTRY_D BIGINT ;
  /* Retrieve the Customer information */

```

```

SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_ID )
= ( SELECT C_BALANCE, C_FIRST, C_MIDDLE , C_ID
    FROM ( SELECT C_ID
           , C_BALANCE
           , C_FIRST
           , C_MIDDLE
           , COUNT(*) OVER() AS COUNT
           , ROWNUMBER() OVER (ORDER BY C_FIRST) AS
NUM
    FROM CUSTOMER
    WHERE C_W_ID = ORD_C_LAST.W_ID
      AND C_D_ID = ORD_C_LAST.D_ID
      AND C_LAST = ORD_C_LAST.C_LAST
    ) AS V1
  WHERE NUM = (COUNT + BIGINT( 1 )) / BIGINT( 2 )
);
/* Take advantage of the index to fetch the first row (and hence
max(o_id) ) */
SET ( O_ID , O_CARRIER_ID , O_ENTRY_D )
= ( SELECT O_ID
    , O_CARRIER_ID
    , O_ENTRY_D
    FROM ORDERS
    WHERE O_W_ID = ORD_C_LAST.W_ID
      AND O_D_ID = ORD_C_LAST.D_ID
      AND O_C_ID = VAR.C_ID
    ORDER BY O_ID DESC
    FETCH FIRST 1 ROW ONLY
    )
;
RETURN VALUES ( VAR.O_ID
                , VAR.O_CARRIER_ID
                , VAR.O_ENTRY_D
                , VAR.C_BALANCE
                , VAR.C_FIRST
                , VAR.C_MIDDLE
                , VAR.C_ID
                )
;
END $
CREATE FUNCTION ORD_C_ID( W_ID INTEGER
                        , D_ID SMALLINT
                        , C_ID INTEGER
                        )
  RETURNS TABLE( O_ID INTEGER
                , O_CARRIER_ID SMALLINT
                , O_ENTRY_D BIGINT
                , C_BALANCE BIGINT
                , C_FIRST VARCHAR(16)
                , C_MIDDLE CHAR(2)
                , C_LAST VARCHAR(16)
                )

```



```

SPECIFIC ORD_C_ID

READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC
LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE C_BALANCE BIGINT ;
DECLARE C_FIRST VARCHAR(16);
DECLARE C_MIDDLE CHAR(2);
DECLARE C_LAST VARCHAR(16);
DECLARE O_ID INTEGER;
DECLARE O_CARRIER_ID SMALLINT;
DECLARE O_ENTRY_D BIGINT;

/* Retrieve the Customer information */

SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_LAST )

= ( SELECT C_BALANCE, C_FIRST, C_MIDDLE, C_LAST

FROM CUSTOMER

WHERE C_ID = ORD_C_ID.C_ID
AND C_W_ID = ORD_C_ID.W_ID
AND C_D_ID = ORD_C_ID.D_ID
)
;

SET (O_ID, O_CARRIER_ID, O_ENTRY_D)

= ( SELECT O_ID
, O_CARRIER_ID
, O_ENTRY_D

FROM ORDERS

WHERE O_W_ID = ORD_C_ID.W_ID
AND O_D_ID = ORD_C_ID.D_ID
AND O_C_ID = ORD_C_ID.C_ID

ORDER BY O_ID DESC
FETCH FIRST 1 ROW ONLY
)
;

RETURN VALUES ( VAR.O_ID
, VAR.O_CARRIER_ID
, VAR.O_ENTRY_D
, VAR.C_BALANCE
, VAR.C_FIRST
, VAR.C_MIDDLE
, VAR.C_LAST
);

END $

--
-- PAYMENT
--

CREATE FUNCTION PAY_C_LAST( W_ID INTEGER
, D_ID SMALLINT
, C_W_ID INTEGER
, C_D_ID SMALLINT
, C_LAST VARCHAR(16)
, H_DATE BIGINT
, H_AMOUNT BIGINT
, BAD_CREDIT_PREFIX VARCHAR(28)
)

```

```

RETURNS TABLE( W_STREET_1 CHAR(20)
, W_STREET_2 CHAR(20)
, W_CITY CHAR(20)
, W_STATE CHAR(2)
, W_ZIP CHAR(9)
, D_STREET_1 CHAR(20)
, D_STREET_2 CHAR(20)
, D_CITY CHAR(20)
, D_STATE CHAR(2)
, D_ZIP CHAR(9)
, C_ID INTEGER
, C_FIRST VARCHAR(16)
, C_MIDDLE CHAR(2)
, C_STREET_1 VARCHAR(20)
, C_STREET_2 VARCHAR(20)
, C_CITY VARCHAR(20)
, C_STATE CHAR(2)
, C_ZIP CHAR(9)
, C_PHONE CHAR(16)
, C_SINCE BIGINT
, C_CREDIT CHAR(2)
, C_CREDIT_LIM BIGINT
, C_DISCOUNT INTEGER
, C_BALANCE BIGINT
, C_DATA CHAR(200)
)

SPECIFIC PAY_C_LAST

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE W_NAME CHAR(10);
DECLARE D_NAME CHAR(10);

DECLARE W_STREET_1 CHAR(20);
DECLARE W_STREET_2 CHAR(20);
DECLARE W_CITY CHAR(20);
DECLARE W_STATE CHAR(2);
DECLARE W_ZIP CHAR(9);

DECLARE D_STREET_1 CHAR(20);
DECLARE D_STREET_2 CHAR(20);
DECLARE D_CITY CHAR(20);
DECLARE D_STATE CHAR(2);
DECLARE D_ZIP CHAR(9);

DECLARE C_ID INTEGER;

DECLARE C_FIRST VARCHAR(16);
DECLARE C_MIDDLE CHAR(2);
DECLARE C_STREET_1 VARCHAR(20);
DECLARE C_STREET_2 VARCHAR(20);
DECLARE C_CITY VARCHAR(20);
DECLARE C_STATE CHAR(2);
DECLARE C_ZIP CHAR(9);
DECLARE C_PHONE CHAR(16);
DECLARE C_SINCE BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);

/* Update District and retrieve its data */

SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP)

```

```

= ( SELECT  D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP
FROM OLD TABLE ( UPDATE DISTRICT
SET D_YTD = D_YTD +
PAY_C_LAST.H_AMOUNT
WHERE D_W_ID = PAY_C_LAST.W_ID
AND D_ID = PAY_C_LAST.D_ID
) AS U
);
/* Determine the C_ID */
SET ( C_ID )
= ( SELECT C_ID
FROM ( SELECT C_ID
, COUNT(*) OVER() AS COUNT
, ROWNUMBER() OVER (ORDER BY C_FIRST) AS
NUM
FROM CUSTOMER
WHERE C_LAST = PAY_C_LAST.C_LAST
AND C_W_ID = PAY_C_LAST.C_W_ID
AND C_D_ID = PAY_C_LAST.C_D_ID
) AS T
WHERE NUM = (COUNT + BIGINT( 1 )) / BIGINT( 2 )
);
/* Update the middle customer */
SET ( C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA )
= ( SELECT C_ID, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE
, CASE WHEN C_CREDIT = 'BC' THEN
SUBSTR(C_DATA, 1, 200) ELSE NULL END AS C_DATA
FROM NEW TABLE ( UPDATE CUSTOMER
SET C_BALANCE = C_BALANCE -
PAY_C_LAST.H_AMOUNT
, C_YTD_PAYMENT = C_YTD_PAYMENT +
PAY_C_LAST.H_AMOUNT
, C_PAYMENT_CNT = C_PAYMENT_CNT +
SMALLINT( 1 )
, C_DATA = CASE WHEN C_CREDIT = 'BC'
THEN CHAR( C_ID ) -- 11
bytes long
|| BAD_CREDIT_PREFIX --
28 bytes long
|| SUBSTR( C_DATA, 1, 461 ) --
461 + 39 = 500
ELSE C_DATA
END
WHERE C_W_ID = PAY_C_LAST.C_W_ID
AND C_D_ID = PAY_C_LAST.C_D_ID
AND C_ID = VAR.C_ID

```

```

) AS U
);
/* Update the warehouse */
SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP )
= ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
FROM OLD TABLE ( UPDATE WAREHOUSE
SET W_YTD = W_YTD +
PAY_C_LAST.H_AMOUNT
WHERE W_ID = PAY_C_LAST.W_ID
) AS U
);
/* Finally insert into the warehouse */
INSERT
INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID,
H_W_ID, H_DATA, H_DATE, H_AMOUNT )
VALUES ( VAR.C_ID
, PAY_C_LAST.C_D_ID
, PAY_C_LAST.C_W_ID
, PAY_C_LAST.D_ID
, PAY_C_LAST.W_ID
, VAR.W_NAME || CHAR( ' ', 4 ) || VAR.D_NAME
, PAY_C_LAST.H_DATE
, PAY_C_LAST.H_AMOUNT
);
/* Done - return the collected data */
RETURN VALUES ( W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP
, D_STREET_1, D_STREET_2, D_CITY, D_STATE,
D_ZIP
, C_ID, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA
);
END
$
CREATE FUNCTION PAY_C_ID( W_ID INTEGER
, D_ID SMALLINT
, C_W_ID INTEGER
, C_D_ID SMALLINT
, C_ID INTEGER
, H_DATE BIGINT
, H_AMOUNT BIGINT
, BAD_CREDIT_PREFIX VARCHAR(34)
)
RETURNS TABLE( W_STREET_1 CHAR(20)
, W_STREET_2 CHAR(20)
, W_CITY CHAR(20)
, W_STATE CHAR(2)
, W_ZIP CHAR(9)

```

```

, D_STREET_1 CHAR(20)
, D_STREET_2 CHAR(20)
, D_CITY CHAR(20)
, D_STATE CHAR(2)
, D_ZIP CHAR(9)
, C_LAST VARCHAR(16)
, C_FIRST VARCHAR(16)
, C_MIDDLE CHAR(2)
, C_STREET_1 VARCHAR(20)
, C_STREET_2 VARCHAR(20)
, C_CITY VARCHAR(20)
, C_STATE CHAR(2)
, C_ZIP CHAR(9)
, C_PHONE CHAR(16)
, C_SINCE BIGINT
, C_CREDIT CHAR(2)
, C_CREDIT_LIM BIGINT
, C_DISCOUNT INTEGER
, C_BALANCE BIGINT
, C_DATA CHAR(200)
)
SPECIFIC PAY_C_ID

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE W_NAME CHAR(10);
DECLARE D_NAME CHAR(10);

DECLARE W_STREET_1 CHAR(20);
DECLARE W_STREET_2 CHAR(20);
DECLARE W_CITY CHAR(20);
DECLARE W_STATE CHAR(2);
DECLARE W_ZIP CHAR(9);

DECLARE D_STREET_1 CHAR(20);
DECLARE D_STREET_2 CHAR(20);
DECLARE D_CITY CHAR(20);
DECLARE D_STATE CHAR(2);
DECLARE D_ZIP CHAR(9);

DECLARE C_LAST VARCHAR(16);

DECLARE C_FIRST VARCHAR(16);
DECLARE C_MIDDLE CHAR(2);
DECLARE C_STREET_1 VARCHAR(20);
DECLARE C_STREET_2 VARCHAR(20);
DECLARE C_CITY VARCHAR(20);
DECLARE C_STATE CHAR(2);
DECLARE C_ZIP CHAR(9);
DECLARE C_PHONE CHAR(16);
DECLARE C_SINCE BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);

/* Update District and retrieve its data */

SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP )

= ( SELECT D_NAME, D_STREET_1, D_STREET_2, D_CITY,
D_STATE, D_ZIP

FROM OLD TABLE ( UPDATE DISTRICT

```

```

SET D_YTD = D_YTD +
PAY_C_ID.H_AMOUNT

WHERE D_W_ID = PAY_C_ID.W_ID
AND D_ID = PAY_C_ID.D_ID
) AS U
)
;

/* Update the middle customer */

SET ( C_LAST, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA )

= ( SELECT C_LAST, C_FIRST, C_MIDDLE, C_STREET_1,
C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE
, CASE WHEN C_CREDIT = 'BC' THEN
SUBSTR(C_DATA, 1, 200) ELSE NULL END AS C_DATA

FROM NEW TABLE ( UPDATE CUSTOMER

SET C_BALANCE = C_BALANCE -
PAY_C_ID.H_AMOUNT
, C_YTD_PAYMENT = C_YTD_PAYMENT +
PAY_C_ID.H_AMOUNT
, C_PAYMENT_CNT = C_PAYMENT_CNT +
SMALLINT( 1 )

, C_DATA = CASE WHEN C_CREDIT = 'BC'
THEN BAD_CREDIT_PREFIX
-- 34 bytes long
|| SUBSTR( C_DATA, 1, 466 ) --
466 + 34 = 500 bytes
ELSE C_DATA
END

WHERE C_W_ID = PAY_C_ID.C_W_ID
AND C_D_ID = PAY_C_ID.C_D_ID
AND C_ID = PAY_C_ID.C_ID
) AS U
)
;

/* Update the warehouse */

SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP )

= ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY,
W_STATE, W_ZIP

FROM OLD TABLE ( UPDATE WAREHOUSE

SET W_YTD = W_YTD + PAY_C_ID.H_AMOUNT

WHERE W_ID = PAY_C_ID.W_ID
) AS U
)
;

/* Finally insert into the warehouse */

INSERT

INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID,
H_W_ID, H_DATA, H_DATE, H_AMOUNT )

```

```

VALUES ( PAY_C_ID.C_ID
        , PAY_C_ID.C_D_ID
        , PAY_C_ID.C_W_ID
        , PAY_C_ID.D_ID
        , PAY_C_ID.W_ID
        , VAR.W_NAME || CHAR(' ', 4) || VAR.D_NAME
        , PAY_C_ID.H_DATE
        , PAY_C_ID.H_AMOUNT
        )
;

/* Done - return the collected data */

RETURN VALUES ( W_STREET_1, W_STREET_2, W_CITY,
                W_STATE, W_ZIP
                , D_STREET_1, D_STREET_2, D_CITY, D_STATE,
                D_ZIP
                , C_LAST, C_FIRST, C_MIDDLE, C_STREET_1,
                C_STREET_2
                , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE,
                C_CREDIT, C_CREDIT_LIM
                , C_DISCOUNT, C_BALANCE, C_DATA
                )
;

END
$

--
-- NEW ORDER
--

CREATE FUNCTION NEW_OL_ALL( I_ID INT
                          , I_QTY SMALLINT
                          , W_ID INT
                          , SUPP_W_ID INT
                          , O_ID INT
                          , D_ID SMALLINT
                          )

RETURNS TABLE( I_PRICE INTEGER
               , I_NAME CHAR(24)
               , I_DATA VARCHAR(50)
               , OL_DIST_INFO CHAR(24)
               , S_DATA VARCHAR(50)
               , S_QUANTITY SMALLINT
               )

SPECIFIC NEW_OL_ALL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE I_PRICE INTEGER ;
DECLARE I_NAME CHAR(24) ;
DECLARE I_DATA VARCHAR(50) ;
DECLARE OL_DIST_INFO CHAR(24) ;
DECLARE S_DATA VARCHAR(50) ;
DECLARE S_QUANTITY SMALLINT ;

SET ( I_PRICE , I_NAME , I_DATA )

= ( SELECT
    I_PRICE
    , I_NAME
    , I_DATA

FROM ITEM

```

```

WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
);

SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )

= ( SELECT OL_DIST_INFO
        , S_DATA
        , S_QUANTITY

FROM NEW TABLE ( UPDATE STOCK

INCLUDE ( OL_DIST_INFO CHAR( 24 ) )

SET S_QUANTITY = CASE WHEN
S_QUANTITY - NEW_OL_ALL.I_QTY >= 10
THEN S_QUANTITY -
NEW_OL_ALL.I_QTY
ELSE S_QUANTITY -
NEW_OL_ALL.I_QTY + 91
END
, S_ORDER_CNT =
S_ORDER_CNT + SMALLINT( 1 )
, S_YTD = S_YTD +
NEW_OL_ALL.I_QTY
, S_REMOTE_CNT = CASE WHEN
NEW_OL_ALL.SUPP_W_ID = NEW_OL_ALL.W_ID
THEN
S_REMOTE_CNT
ELSE
S_REMOTE_CNT + SMALLINT( 1 )
END
, OL_DIST_INFO = CASE D_ID
WHEN SMALLINT( 1 ) THEN S_DIST_01
WHEN
SMALLINT( 2 ) THEN S_DIST_02
WHEN
SMALLINT( 3 ) THEN S_DIST_03
WHEN
SMALLINT( 4 ) THEN S_DIST_04
WHEN
SMALLINT( 5 ) THEN S_DIST_05
WHEN
SMALLINT( 6 ) THEN S_DIST_06
WHEN
SMALLINT( 7 ) THEN S_DIST_07
WHEN
SMALLINT( 8 ) THEN S_DIST_08
WHEN
SMALLINT( 9 ) THEN S_DIST_09
WHEN
SMALLINT( 10 ) THEN S_DIST_10
END
WHERE S_I_ID = NEW_OL_ALL.I_ID
AND S_W_ID = NEW_OL_ALL.SUPP_W_ID
) AS U
)
;

RETURN VALUES( VAR.I_PRICE
                , VAR.I_NAME
                , VAR.I_DATA
                , VAR.OL_DIST_INFO
                , VAR.S_DATA
                , VAR.S_QUANTITY
                )
;

```

```

END
$

CREATE FUNCTION NEW_OL_LOCAL( I_ID INT
    , I_QTY SMALLINT
    , W_ID INT
    , O_ID INT
    , D_ID SMALLINT
    )

RETURNS TABLE( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , OL_DIST_INFO CHAR(24)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT
    )

SPECIFIC NEW_OL_LOCAL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE I_PRICE INTEGER ;
DECLARE I_NAME CHAR(24) ;
DECLARE I_DATA VARCHAR(50) ;
DECLARE OL_DIST_INFO CHAR(24) ;
DECLARE S_DATA VARCHAR(50) ;
DECLARE S_QUANTITY SMALLINT ;

SET ( I_PRICE , I_NAME , I_DATA )

= ( SELECT
    I_PRICE
    , I_NAME
    , I_DATA

    FROM ITEM

    WHERE ITEM.I_ID = NEW_OL_LOCAL.I_ID
    );

SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )

= ( SELECT OL_DIST_INFO
    , S_DATA
    , S_QUANTITY

    FROM NEW TABLE ( UPDATE STOCK

        INCLUDE ( OL_DIST_INFO CHAR( 24 ) )

        SET S_QUANTITY = CASE WHEN
S_QUANTITY - NEW_OL_LOCAL.I_QTY >= 10
        THEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY
        ELSE S_QUANTITY -
NEW_OL_LOCAL.I_QTY + 91
        END
    , S_ORDER_CNT =
S_ORDER_CNT + SMALLINT( 1 )
    , S_YTD = S_YTD +
NEW_OL_LOCAL.I_QTY
    , OL_DIST_INFO = CASE D_ID
WHEN SMALLINT( 1 ) THEN S_DIST_01

```

```

SMALLINT( 2 ) THEN S_DIST_02
SMALLINT( 3 ) THEN S_DIST_03
SMALLINT( 4 ) THEN S_DIST_04
SMALLINT( 5 ) THEN S_DIST_05
SMALLINT( 6 ) THEN S_DIST_06
SMALLINT( 7 ) THEN S_DIST_07
SMALLINT( 8 ) THEN S_DIST_08
SMALLINT( 9 ) THEN S_DIST_09
SMALLINT( 10 ) THEN S_DIST_10
    )
    END
    WHERE S_I_ID = NEW_OL_LOCAL.I_ID
    AND S_W_ID = NEW_OL_LOCAL.W_ID
    ) AS U
    )
    ;

RETURN VALUES( VAR.I_PRICE
    , VAR.I_NAME
    , VAR.I_DATA
    , VAR.OL_DIST_INFO
    , VAR.S_DATA
    , VAR.S_QUANTITY
    )
    ;

END
$

CREATE FUNCTION NEW_WH ( O_ID INTEGER
    , W_ID INTEGER
    , D_ID SMALLINT
    , C_ID INTEGER
    , O_ENTRY_D BIGINT
    , O_OL_CNT SMALLINT
    , O_ALL_LOCAL SMALLINT
    )

RETURNS TABLE ( W_TAX INTEGER
    , C_DISCOUNT INTEGER
    , C_LAST VARCHAR(16)
    , C_CREDIT CHAR(2)
    )

SPECIFIC NEW_WH

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION
LANGUAGE SQL

VAR: BEGIN ATOMIC

DECLARE C_DISCOUNT INTEGER ;
DECLARE C_LAST VARCHAR(16) ;
DECLARE C_CREDIT CHAR(2) ;
DECLARE W_TAX INTEGER ;

INSERT

    INTO NEW_ORDER ( NO_O_ID, NO_D_ID, NO_W_ID )

    VALUES ( O_ID
    , D_ID
    , W_ID

```

```

)
;

INSERT

  INTO ORDERS ( O_C_ID, O_ENTRY_D, O_CARRIER_ID,
O_OL_CNT, O_ALL_LOCAL, O_ID, O_W_ID, O_D_ID )

  VALUES ( C_ID
           , O_ENTRY_D
           , 0
           , O_OL_CNT
           , O_ALL_LOCAL
           , O_ID
           , W_ID
           , D_ID
           )
;

SET ( C_DISCOUNT, C_LAST, C_CREDIT )
= ( SELECT C_DISCOUNT, C_LAST, C_CREDIT

    FROM CUSTOMER

    WHERE C_ID = NEW_WH.C_ID
      AND C_W_ID = W_ID
      AND C_D_ID = D_ID
    )
;

SET W_TAX
= ( SELECT W_TAX

    FROM WAREHOUSE

    WHERE W_ID = NEW_WH.W_ID
    )
;

RETURN VALUES ( W_TAX , C_DISCOUNT , C_LAST ,
C_CREDIT ) ;

END
$

```

Src.Srv/cat-proc.ddl

```

CREATE PROCEDURE news
  (in new_in  varchar(270) FOR BIT DATA,
  out new_out varchar(662) FOR BIT DATA)
  LANGUAGE C
  PARAMETER STYLE GENERAL
  EXTERNAL NAME
'/home/tpcc/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/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_newword->s_W_ID
#define d_id in_newword->s_D_ID
#define c_id in_newword->s_C_ID
#define o_entry_d in_newword->s_O_ENTRY_D_time

#define inputItemCount in_newword->s_O_OL_CNT

#define allLocal in_newword->s_all_local

```

```

// Redirected output fields

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

// This output field becomes an input field to order_line

#define next_o_id newword->s_O_ID

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

#define ol_quantity0 in_newword->in_item [ 0 ].s_OL_QUANTITY
#define ol_quantity1 in_newword->in_item [ 1 ].s_OL_QUANTITY
#define ol_quantity2 in_newword->in_item [ 2 ].s_OL_QUANTITY
#define ol_quantity3 in_newword->in_item [ 3 ].s_OL_QUANTITY
#define ol_quantity4 in_newword->in_item [ 4 ].s_OL_QUANTITY
#define ol_quantity5 in_newword->in_item [ 5 ].s_OL_QUANTITY
#define ol_quantity6 in_newword->in_item [ 6 ].s_OL_QUANTITY
#define ol_quantity7 in_newword->in_item [ 7 ].s_OL_QUANTITY
#define ol_quantity8 in_newword->in_item [ 8 ].s_OL_QUANTITY
#define ol_quantity9 in_newword->in_item [ 9 ].s_OL_QUANTITY
#define ol_quantity10 in_newword->in_item [ 10 ].s_OL_QUANTITY
#define ol_quantity11 in_newword->in_item [ 11 ].s_OL_QUANTITY
#define ol_quantity12 in_newword->in_item [ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_newword->in_item [ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_newword->in_item [ 14 ].s_OL_QUANTITY

#define supply_w_id0 in_newword-
>in_item [ 0 ].s_OL_SUPPLY_W_ID

```

```

#define supply_w_id1 in_neword-
>in_item[ 1 ].s_OL_SUPPLY_W_ID
#define supply_w_id2 in_neword-
>in_item[ 2 ].s_OL_SUPPLY_W_ID
#define supply_w_id3 in_neword-
>in_item[ 3 ].s_OL_SUPPLY_W_ID
#define supply_w_id4 in_neword-
>in_item[ 4 ].s_OL_SUPPLY_W_ID
#define supply_w_id5 in_neword-
>in_item[ 5 ].s_OL_SUPPLY_W_ID
#define supply_w_id6 in_neword-
>in_item[ 6 ].s_OL_SUPPLY_W_ID
#define supply_w_id7 in_neword-
>in_item[ 7 ].s_OL_SUPPLY_W_ID
#define supply_w_id8 in_neword-
>in_item[ 8 ].s_OL_SUPPLY_W_ID
#define supply_w_id9 in_neword-
>in_item[ 9 ].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_neword-
>in_item[ 10 ].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_neword-
>in_item[ 11 ].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_neword-
>in_item[ 12 ].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_neword-
>in_item[ 13 ].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_neword-
>in_item[ 14 ].s_OL_SUPPLY_W_ID

```

```
EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR
```

```

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

```

```

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

```

```
FROM Table( VALUES
```

```

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

```

```

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

```

```
WHERE NEW_OL_ALL.I_PRICE IS NOT
```

```
NULL
```

```
)
```

```
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
```

```
FROM NEW TABLE ( INSERT INTO ORDER_LINE
```

```

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

```

```

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

```

```

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY

```

```

, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,

```

```
S_QUANTITY
```

```
FROM DATA
```

```
) AS INS
```

```
;
```

```
EXEC SQL DECLARE ISOL_Remote_2 CURSOR FOR
```

```

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

```

```

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

```

```
FROM Table( VALUES
```

```
( SMALLINT( 1) , :id0 , :ol_quantity0 , :supply_w_id0 )
```

```
( SMALLINT( 2) , :id1 , :ol_quantity1 , :supply_w_id1 )
```



```

        ) AS X ( OL_NUMBER , I_ID ,
        I_QTY , I_SUPPLY_W_ID )
        ) AS ITEMLIST
        , TABLE( NEW_OL_ALL( I_ID
        , I_QTY
        , W_ID
        , I_SUPPLY_W_ID
        , O_ID
        , D_ID
        )
        ) AS NEW_OL_ALL
        WHERE NEW_OL_ALL.I_PRICE IS NOT
        NULL
    )

```

```

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
    S_DATA , S_QUANTITY

```

```

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

```

```

    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )

```

```

    INCLUDE ( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )

```

```

    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,

```

```

S_QUANTITY

```

```

    FROM DATA

```

```

    ) AS INS

```

```

;

```

```

EXEC SQL DECLARE ISOL_Remote_3 CURSOR FOR

```

```

    WITH DATA AS ( SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , 0 AS OL_DELIVERY_D
    , I_QTY
    , (I_PRICE * I_QTY) AS TOTAL_PRICE
    , OL_DIST_INFO

```

```

, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

```

```

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

```

```

    FROM Table( VALUES

```

```

    ( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )

```

```

    ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )

```

```

    ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )

```

```

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

```

```

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

```

```

        WHERE NEW_OL_ALL.I_PRICE IS NOT

```

```

        NULL

```

```

    )

```

```

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
    S_DATA , S_QUANTITY

```

```

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

```

```

    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )

```

```

    INCLUDE ( I_PRICE INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )

```

```

    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,

```

```

S_QUANTITY

```

```

FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
, ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
, ( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO

```

```

)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_5 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
, ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
, ( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
, ( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
)

```

```

        ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE 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
)
) 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
)
) 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
                        )
        ) 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
                        )
        ) 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
)
) AS NEW_OL_ALL

```

```

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_11 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )

```

```

( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO

```

```

, I_PRICE , I_NAME , I_DATA , S_DATA ,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_12 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 , :supply_w_id11 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL

```

```

)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_13 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )

```

```

, :id2 , :ol_quantity2 , :supply_w_id2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 , :supply_w_id9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 , :supply_w_id10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 , :supply_w_id11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 , :supply_w_id12 )

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

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

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL

)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY

```



```

, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_14 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 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 )

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

, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID

```

```

, D_ID
)
) AS NEW_OL_ALL

WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 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 )
I_QTY , I_SUPPLY_W_ID ) AS X ( OL_NUMBER , I_ID ,
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)

```

```

, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_1 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID

```

```

, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_2 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)

```

```

) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_3 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
, ( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)

```

```

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )

) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID

```

```

, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID

```

```

        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

        FROM DATA

        ) AS INS
;

EXEC SQL DECLARE ISOL_Local_5 CURSOR FOR

WITH DATA AS ( SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , W_ID AS I_SUPPLY_W_ID

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

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

                FROM Table( VALUES

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

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

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

        WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER

```

```

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

INCLUDE ( I_PRICE INTEGER
        , I_NAME CHAR(24)
        , I_DATA VARCHAR(50)
        , S_DATA VARCHAR(50)
        , S_QUANTITY SMALLINT )

SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

        FROM DATA

        ) AS INS
;

EXEC SQL DECLARE ISOL_Local_6 CURSOR FOR

WITH DATA AS ( SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , W_ID AS I_SUPPLY_W_ID
        , 0 AS OL_DELIVERY_D
        , I_QTY
        , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

                FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )

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

```

```

        , TABLE( NEW_OL_LOCAL( I_ID
                        , I_QTY
                        , W_ID
                        , O_ID
                        , D_ID
                        )
                ) AS NEW_OL_LOCAL
        WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
    )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

        ( OL_O_ID
        , OL_D_ID
        , OL_W_ID
        , OL_NUMBER
        , OL_I_ID
        , OL_SUPPLY_W_ID
        , OL_DELIVERY_D
        , OL_QUANTITY
        , OL_AMOUNT
        , OL_DIST_INFO
        )

        INCLUDE ( I_PRICE INTEGER
        , I_NAME CHAR(24)
        , I_DATA VARCHAR(50)
        , S_DATA VARCHAR(50)
        , S_QUANTITY SMALLINT )

        SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

        FROM DATA

        ) AS INS
;

EXEC SQL DECLARE ISOL_Local_7 CURSOR FOR

WITH DATA AS ( SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , W_ID AS I_SUPPLY_W_ID
        , 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 )

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

        , TABLE( NEW_OL_LOCAL( I_ID
                        , I_QTY
                        , W_ID
                        , O_ID
                        , D_ID
                        )
                ) AS NEW_OL_LOCAL
        WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
    )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

        ( OL_O_ID
        , OL_D_ID
        , OL_W_ID
        , OL_NUMBER
        , OL_I_ID
        , OL_SUPPLY_W_ID
        , OL_DELIVERY_D
        , OL_QUANTITY
        , OL_AMOUNT
        , OL_DIST_INFO
        )

        INCLUDE ( I_PRICE INTEGER
        , I_NAME CHAR(24)
        , I_DATA VARCHAR(50)
        , S_DATA VARCHAR(50)
        , S_QUANTITY SMALLINT )

        SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

```

```

FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_8 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 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
)
) 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
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE

```

```

, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

```



```

SELECT O_ID
      , D_ID
      , W_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , OL_DELIVERY_D
      , I_QTY
      , TOTAL_PRICE
      , OL_DIST_INFO
      , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_11 CURSOR FOR

WITH DATA AS ( SELECT O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , W_ID AS I_SUPPLY_W_ID
                  , 0 AS OL_DELIVERY_D
                  , I_QTY
                  , (I_PRICE * I_QTY) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )

) AS X ( OL_NUMBER , I_ID ,
I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
                     , I_QTY
                     , W_ID

```

```

      , O_ID
      , D_ID
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
  , OL_D_ID
  , OL_W_ID
  , OL_NUMBER
  , OL_I_ID
  , OL_SUPPLY_W_ID
  , OL_DELIVERY_D
  , OL_QUANTITY
  , OL_AMOUNT
  , OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
          , I_NAME CHAR(24)
          , I_DATA VARCHAR(50)
          , S_DATA VARCHAR(50)
          , S_QUANTITY SMALLINT )

SELECT O_ID
      , D_ID
      , W_ID
      , OL_NUMBER
      , I_ID
      , I_SUPPLY_W_ID
      , OL_DELIVERY_D
      , I_QTY
      , TOTAL_PRICE
      , OL_DIST_INFO
      , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_12 CURSOR FOR

WITH DATA AS ( SELECT O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , W_ID AS I_SUPPLY_W_ID
                  , 0 AS OL_DELIVERY_D
                  , I_QTY
                  , (I_PRICE * I_QTY) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

FROM Table( VALUES

```

```

FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )

) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER

```

```

, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_13 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
) AS X ( OL_NUMBER , I_ID ,
I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID

```

```

        , D_ID
      )
    ) AS NEW_OL_LOCAL
  WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_14 CURSOR FOR

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

FROM Table( VALUES

```

```

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
,
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
,
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
,
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
,
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
,
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
,
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
,
( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
,
( SMALLINT( 14 ) , :id13 , :ol_quantity13 )

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

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

WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

FROM NEW TABLE ( INSERT INTO ORDER_LINE

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)

INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )

SELECT O_ID
, D_ID

```

```

        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

        FROM DATA

        ) AS INS
;

EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

WITH DATA AS ( SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , W_ID AS I_SUPPLY_W_ID
        , 0 AS OL_DELIVERY_D
        , I_QTY
        , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

        FROM Table( VALUES

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
        ,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
        ,
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
        ,
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
        ,
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
        ,
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
        ,
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
        ,
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
        ,
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
        ,
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
        ,
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
        ,
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
        ,
( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
        ,
( SMALLINT( 14 ) , :id13 , :ol_quantity13 )
        ,
( SMALLINT( 15 ) , :id14 , :ol_quantity14 )

        ) AS X ( OL_NUMBER , I_ID ,
I_QTY

```

```

        ) AS ITEMLIST

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

        WHERE NEW_OL_LOCAL.I_PRICE IS NOT
NULL
    )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

        ( OL_O_ID
        , OL_D_ID
        , OL_W_ID
        , OL_NUMBER
        , OL_I_ID
        , OL_SUPPLY_W_ID
        , OL_DELIVERY_D
        , OL_QUANTITY
        , OL_AMOUNT
        , OL_DIST_INFO
        )

    INCLUDE ( I_PRICE INTEGER
        , I_NAME CHAR(24)
        , I_DATA VARCHAR(50)
        , S_DATA VARCHAR(50)
        , S_QUANTITY SMALLINT )

    SELECT O_ID
        , D_ID
        , W_ID
        , OL_NUMBER
        , I_ID
        , I_SUPPLY_W_ID
        , OL_DELIVERY_D
        , I_QTY
        , TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

        FROM DATA

        ) AS INS
;

// Start processing

in_newword = (struct in_newword_struct *) pin ;
newword = (struct out_newword_struct *) pout ;

#ifdef DEBUGIT
    new_debug( newword, in_newword, "SP upon entry");
#endif

// Using I_PRICE == 0 as a flag to the client that the ITEM was not
// fetched (hence bad).

for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < in_newword-
>s_O_OL_CNT ; inputItemArrayIndex++)
{

```

```

    i_priceArray[ inputItemArrayIndex ] = 0 ;
}

neword->deadlocks = -1 ;

retry_tran:

neword->deadlocks++ ;

EXEC SQL

    SELECT D_TAX, D_NEXT_O_ID INTO :dist_tax , :next_o_id

    FROM OLD TABLE ( UPDATE DISTRICT

        SET D_NEXT_O_ID = D_NEXT_O_ID + 1

        WHERE D_W_ID = :w_id
            AND D_ID = :d_id
    ) AS OT
;

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

// 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 )
    {
        neword->s_OL_CNT ++ ;
    }
    else
    if( sqlca.sqlcode == +100 )
    {
        break ;
    }
    else
    goto sql_error ;
}

if ( allLocal )
{
    switch( inputItemCount )
    {
        case 1:
            EXEC SQL OPEN ISOL_Local_1 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;

```

```

            NEW_CURSOR_ERROR
            }
            break ;
        case 2:
            EXEC SQL OPEN ISOL_Local_2 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 3:
            EXEC SQL OPEN ISOL_Local_3 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 4:
            EXEC SQL OPEN ISOL_Local_4 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 5:
            EXEC SQL OPEN ISOL_Local_5 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 6:
            EXEC SQL OPEN ISOL_Local_6 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 7:
            EXEC SQL OPEN ISOL_Local_7 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Local_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_
data , :s_quantity ;

```

```

        NEW_CURSOR_ERROR
    }
    break ;
case 8:
    EXEC SQL OPEN ISOL_Local_8 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 9:
    EXEC SQL OPEN ISOL_Local_9 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 10:
    EXEC SQL OPEN ISOL_Local_10 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 11:
    EXEC SQL OPEN ISOL_Local_11 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 12:
    EXEC SQL OPEN ISOL_Local_12 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 13:
    EXEC SQL OPEN ISOL_Local_13 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;

```

```

        NEW_CURSOR_ERROR
    }
    break ;
case 14:
    EXEC SQL OPEN ISOL_Local_14 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 15:
    EXEC SQL OPEN ISOL_Local_15 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;

    default:
        sqlerror(NEWORD_SQL, "Default switch on local
orderline/stock/index", __FILE__, __LINE__, &sqlca);
        goto ferror;
    }
}
else
{
    switch( inputItemCount )
    {
        case 1:
            EXEC SQL OPEN ISOL_Remote_1 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 2:
            EXEC SQL OPEN ISOL_Remote_2 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
                NEW_CURSOR_ERROR
            }
            break ;
        case 3:
            EXEC SQL OPEN ISOL_Remote_3 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
            {
                EXEC SQL FETCH ISOL_Remote_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
                NEW_CURSOR_ERROR
            }

```

```

    }
    break ;
case 4:
    EXEC SQL OPEN ISOL_Remote_4 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 5:
    EXEC SQL OPEN ISOL_Remote_5 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 6:
    EXEC SQL OPEN ISOL_Remote_6 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 7:
    EXEC SQL OPEN ISOL_Remote_7 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 8:
    EXEC SQL OPEN ISOL_Remote_8 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 9:
    EXEC SQL OPEN ISOL_Remote_9 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR

```

```

    }
    break ;
case 10:
    EXEC SQL OPEN ISOL_Remote_10 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 11:
    EXEC SQL OPEN ISOL_Remote_11 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 12:
    EXEC SQL OPEN ISOL_Remote_12 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 13:
    EXEC SQL OPEN ISOL_Remote_13 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 14:
    EXEC SQL OPEN ISOL_Remote_14 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 15:
    EXEC SQL OPEN ISOL_Remote_15 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Remote_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation, :s_
data, :s_quantity ;
        NEW_CURSOR_ERROR

```

```

    }
    break ;

    default:
        sqlerror(NEWORD_SQL, "Default switch on remote
orderline/stock/index", __FILE__, __LINE__, &sqlca);
        goto ferror;
    }
}

for ( inputItemArrayIndex = 0 ;
inputItemArrayIndex < in_neword->s_O_OL_CNT // from
input
&& i_priceArray[ inputItemArrayIndex ] != 0 ;
inputItemArrayIndex++ )
{
    // s_I_NAME, and s_S_QUANTITY already set as output host
variables

    neword->item[ inputItemArrayIndex ].s_I_PRICE =
i_priceArray[ inputItemArrayIndex ] ;

    if ( is_ORIGINAL( s_dataArray[ inputItemArrayIndex ].data,
s_dataArray[ inputItemArrayIndex ].len )
&& is_ORIGINAL( i_dataArray[ inputItemArrayIndex ].data,
i_dataArray[ inputItemArrayIndex ].len ) )
    {
        neword->item[ inputItemArrayIndex ].s_brand_generic = 'B';
    }
    else
    {
        neword->item[ inputItemArrayIndex ].s_brand_generic = 'G';
    }
}

EXEC SQL

SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT

INTO :ware_tax, :c_discount, :c_last, :c_credit

FROM TABLE ( NEW_WH ( :next_o_id
, :w_id
, :d_id
, :c_id
, :o_entry_d
, :inputItemCount
, :allLocal
)
) AS NEW_WH_TABLE
;

if ( sqlca.sqlcode == 0 )
{
    if ( neword->s_O_OL_CNT == in_neword->s_O_OL_CNT )
    {
        neword->s_transtatus = TRAN_OK ;

        EXEC SQL COMMIT;

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

        EXEC SQL ROLLBACK WORK ;
    }
}

```

```

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

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

    sqlerror( NEWORD_SQL, "NEW_WH", __FILE__, __LINE__,
&sqlca);
    goto ferror;
}

/*-----*/
/* Return to client */
/*-----*/

mexit:

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

#ifdef DEBUGIT
    new_debug( neword, in_neword, "SP prior to return");
#endif

return ( storedProcRc ) ;

sql_error:

{
    char tempstr[ 4096 ] ;

    DLCHK( retry_tran ) ;

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

ferror:

neword->s_transtatus = FATAL_SQLERROR;

EXEC SQL ROLLBACK WORK;

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

goto mexit ;
}

/*

```



```

** A little function to search for the string "ORIGINAL" given a string
and
** it's length
*/
static unsigned char skip[256] = {8,8,8,8,8,8,8,8, /*0-9*/
    8,8,8,8,8,8,8,8, /*10-19*/
    8,8,8,8,8,8,8,8, /*20-29*/
    8,8,8,8,8,8,8,8, /*30-39*/
    8,8,8,8,8,8,8,8, /*40-49*/
    8,8,8,8,8,8,8,8, /*50-59*/
    8,8,8,8,8,1,8,8,8, /*60-69*/
    8,4,8,3,8,8,0,8,2,7, /*70-79*/
    8,8,6,8,8,8,8,8,8,8, /*80-89*/
    8,8,8,8,8,8,8,8,8, /*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;

    // NOTE: this varchar would normally live inside the declare
section
    // but this package already declared the same field higher up.
Need the field
    // within this scope though.

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

    int storedProcRc ;
    int itemArrayIndex = 0 ;

#define w_id      in_ordstat->s_W_ID ;
#define d_id      in_ordstat->s_D_ID ;
#define c_id_input  in_ordstat->s_C_ID
#define o_id      ordstat->s_O_ID

```

```

#define o_entry_d      ordstat->s_O_ENTRY_D_time
#define o_carrier_id  ordstat->s_O_CARRIER_ID
#define c_id          ordstat->s_C_ID
#define c_first       ordstat->s_C_FIRST
#define c_middle      ordstat->s_C_MIDDLE
#define c_last        ordstat->s_C_LAST
#define c_balance     ordstat->s_C_BALANCE

EXEC SQL DECLARE read_orderline_cur CURSOR FOR

    SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY,
    OL_AMOUNT, OL_DELIVERY_D

    FROM ORDER_LINE

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

    FOR FETCH ONLY ;

ordstat->deadlocks = -1 ;

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

retry_tran:

ordstat->deadlocks ++;

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

EXEC SQL

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

INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_middl
e, :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_middl
e, :c_last

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

```

```

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

/*-----*/
/* Read ORDER_LINES */
/*-----*/

EXEC SQL OPEN read_orderline_cur ;

if ( sqlca.sqlcode != 0 )
{
DLCHK( retry_tran );
sqlerror(ORDSTAT_SQL, "OPEN CURSOR
read_orderline_cur", __FILE__, __LINE__, &sqlca );
goto ferror;
}

itemArrayIndex = 0 ;
{
do
{
EXEC SQL FETCH read_orderline_cur

INTO :ol_i_id , :ol_supply_w_id , :ol_quantity , :ol_amount , :ol_deliv
ery_d ;

if ( sqlca.sqlcode == 0 )
{
ordstat->item[ itemArrayIndex ].s_OL_I_ID = ol_i_id ;
ordstat->item[ itemArrayIndex ].s_OL_SUPPLY_W_ID =
ol_supply_w_id ;
ordstat->item[ itemArrayIndex ].s_OL_QUANTITY =
ol_quantity ;
ordstat->item[ itemArrayIndex ].s_OL_AMOUNT =
ol_amount ;
ordstat->item[ itemArrayIndex ].s_OL_DELIVERY_D_time =
ol_delivery_d ;

itemArrayIndex++;
}
else
if ( sqlca.sqlcode < 0 )
{
DLCHK( retry_tran );
sqlerror( ORDSTAT_SQL, "FETCH CURSOR
read_orderline_cur", __FILE__, __LINE__, &sqlca );
goto ferror ;
}
}
while ( sqlca.sqlcode == 0 ) ;
}

ordstat->s_ol_cnt = itemArrayIndex ;

EXEC SQL COMMIT ;

if ( sqlca.sqlcode == 0 )
{
ordstat->s_transtatus = TRAN_OK ;
}
else
{
DLCHK( retry_tran );
sqlerror(ORDSTAT_SQL, "COMMIT", __FILE__, __LINE__,
&sqlca);
goto ferror ;
}

```

```

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

#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "SP prior to return");
#endif

    return ( storedProcRc ) ;

ferror:
    ordstat->s_transtatus = FATAL_SQLERROR ;

    EXEC SQL ROLLBACK WORK ;

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

    goto mexit;
}

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

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

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

    struct sqlca sqlca ;

    int storedProcRc ;

    short district_id ;
    sqlint32 customer_id ;

    EXEC SQL BEGIN DECLARE SECTION;

    // input

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

    // output

    short no_o_id_indicator = 0 ;

```

```

    sqlint32 no_o_id ;

    EXEC SQL END DECLARE SECTION;

#define d_id        district_id
#define c_id        customer_id

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

    delivery->deadlocks = -1 ;

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

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

    d_id = 1;

retry_tran:
    delivery->deadlocks++;

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

        EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

            SELECT O_ID

                INTO :no_o_id :no_o_id_indicator

            FROM TABLE

            ( DEL( :w_id , :d_id , :o_carrier_id , :ol_delivery_d ) ) AS T ;

        COMMIT ;

        END COMPOUND ;

        if ( sqlca.sqlcode == 0 )
        {
            /* Refer to clause 2.7.4.2, bullet 3 in spec.*/
            /* Need to report if more than 1 or 1% of */

```

```

/* no_o_id will remain 0 if null returned, so just treat the same
way */

    delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
}
else
{
    DLCHK( retry_tran );

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

delivery->s_transtatus = TRAN_OK ;

mexit:

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

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

return ( storedProcRc ) ;

ferror:

delivery->s_transtatus = FATAL_SQLERROR ;

EXEC SQL ROLLBACK WORK ;

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

goto mexit ;
}

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

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

SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{
    return order_status_internal( pin, pout ) ;
}

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

```

Src.Srv/uncat_func.ddl

```

-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 -
-- 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM
-- Corp.
-----
--
-- uncat-func.ddl - Drop table function DDL
--
-- STOCK LEVEL
DROP SPECIFIC FUNCTION STOCK_LEVEL $
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY $
-- ORDER STATUS
DROP SPECIFIC FUNCTION ORD_C_LAST $
DROP SPECIFIC FUNCTION ORD_C_ID $
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST $
DROP SPECIFIC FUNCTION PAY_C_ID $
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL $
DROP SPECIFIC FUNCTION NEW_OL_LOCAL $
DROP SPECIFIC FUNCTION NEW_WH $

```

Src.Srv/uncat_proc.ddl

```

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

DROP PROCEDURE news;

DROP PROCEDURE pays;

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

DROP PROCEDURE ords;

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

DROP PROCEDURE dels;

DROP PROCEDURE stks
    (varchar(18),varchar(14));

DROP PROCEDURE stks;

```

utils/EXPLAIN.ddl

```

-- *- sql *-
--
-- Sample DDL to create Explain tables for Version 5.0
--
-- -> assumes db2start issued
-- -> assumes connection to a database exists
-- -> assumes called by "db2 -tf EXPLAIN.DDL"
--

```

```

-- To remind users how to use this file!
--
ECHO          ;
ECHO ***** IMPORTANT ***** ;
ECHO          ;
ECHO USAGE: db2 -tf EXPLAIN.DDL ;
ECHO          ;
ECHO ***** IMPORTANT ***** ;
ECHO          ;
ECHO          ;
--
-- Set autocommit off
--
UPDATE COMMAND OPTIONS USING C OFF;
--
-- EXPLAIN INSTANCE
--
-- (must be defined first due to referential integrity defintiions)
--
CREATE TABLE EXPLAIN_INSTANCE ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
                                EXPLAIN_TIME  TIMESTAMP NOT NULL,
                                SOURCE_NAME    VARCHAR(128) NOT
NULL,
                                SOURCE_SCHEMA  VARCHAR(128) NOT
NULL,
                                SOURCE_VERSION VARCHAR(64) NOT
NULL,
                                EXPLAIN_OPTION CHAR(1) NOT NULL,
                                SNAPSHOT_TAKEN CHAR(1) NOT NULL,
                                DB2_VERSION   CHAR(7) NOT NULL,
                                SQL_TYPE      CHAR(1) NOT NULL,
                                QUERYOPT      INTEGER NOT NULL,
                                BLOCK         CHAR(1) NOT NULL,
                                ISOLATION     CHAR(2) NOT NULL,
                                BUFFPAGE     INTEGER NOT NULL,
                                AVG_APPLS    INTEGER NOT NULL,
                                SORTHEAP     INTEGER NOT NULL,
                                LOCKLIST     INTEGER NOT NULL,
                                MAXLOCKS     SMALLINT NOT NULL,
                                LOCKS_AVAIL  INTEGER NOT NULL,
                                CPU_SPEED    DOUBLE NOT NULL,
                                REMARKS      VARCHAR(254),
                                DBHEAP      INTEGER NOT NULL,
                                COMM_SPEED   DOUBLE NOT NULL,
                                PARALLELISM CHAR(2) NOT NULL,
                                DATAJOINER CHAR(1) NOT NULL,
                                PRIMARY KEY (EXPLAIN_REQUESTER,
                                             EXPLAIN_TIME,
                                             SOURCE_NAME,
                                             SOURCE_SCHEMA,
                                             SOURCE_VERSION))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STATEMENT
--
CREATE TABLE EXPLAIN_STATEMENT
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME      TIMESTAMP NOT NULL,
  SOURCE_NAME       VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA     VARCHAR(128) NOT
NULL,
  SOURCE_VERSION    VARCHAR(64) NOT
NULL,
  EXPLAIN_LEVEL    CHAR(1) NOT NULL,
  STMTNO           INTEGER NOT NULL,
  SECTNO           INTEGER NOT NULL,
  QUERYNO          INTEGER NOT NULL,
  QUERYTAG        CHAR(20) NOT NULL,

```

```

STATEMENT_TYPE CHAR(2) NOT
NULL,
UPDATABLE CHAR(1) NOT NULL,
DELETABLE CHAR(1) NOT NULL,
TOTAL_COST DOUBLE NOT NULL,
STATEMENT_TEXT CLOB(2M) NOT
NULL NOT LOGGED,
SNAPSHOT BLOB(10M) NOT
LOGGED,
QUERY_DEGREE INTEGER NOT
NULL,
PRIMARY KEY (EXPLAIN_REQUESTER,
             EXPLAIN_TIME,
             SOURCE_NAME,
             SOURCE_SCHEMA,
             SOURCE_VERSION,
             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,
EXTENTSIZE      INTEGER  NOT NULL,
CLUSTER         DOUBLE  NOT NULL,
NLEAF           INTEGER  NOT NULL,
NLEVELS         INTEGER  NOT NULL,
FULLKEYCARD     BIGINT  NOT NULL,
OVERFLOW        INTEGER  NOT NULL,
FIRSTKEYCARD    BIGINT  NOT NULL,
FIRST2KEYCARD   BIGINT  NOT NULL,
FIRST3KEYCARD   BIGINT  NOT NULL,
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_INSTANCE
--
-- (must be defined first due to referential integrity defintiions)
--
CREATE TABLE ADVISE_INSTANCE (
START_TIME    TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
END_TIME      TIMESTAMP NOT NULL WITH DEFAULT
CURRENT TIMESTAMP,
MODE          VARCHAR(4) NOT NULL WITH DEFAULT "",
WKLD_COMPRESSION CHAR(4) NOT NULL WITH
DEFAULT 'NONE',
STATUS        CHAR(9) NOT NULL WITH DEFAULT "",
PRIMARY KEY (START_TIME))
IN USERSPACE1
INDEX IN USERSPACE1;

```

```

--
-- ADVISE_INDEX
--
CREATE TABLE ADVISE_INDEX(
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL WITH
DEFAULT "",
EXPLAIN_TIME    TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
SOURCE_NAME     VARCHAR(128) NOT NULL WITH
DEFAULT "",
SOURCE_SCHEMA   VARCHAR(128) NOT NULL WITH
DEFAULT "",
SOURCE_VERSION   VARCHAR(64) NOT NULL WITH
DEFAULT "",
EXPLAIN_LEVEL   CHAR(1) NOT NULL WITH DEFAULT
",
STMTNO          INTEGER NOT NULL WITH DEFAULT 0,
SECTNO          INTEGER NOT NULL WITH DEFAULT 0,
QUERYNO         INTEGER NOT NULL WITH DEFAULT
0,
QUERYTAG        CHAR(20) NOT NULL WITH DEFAULT
",
NAME            VARCHAR(128) NOT NULL,
CREATOR         VARCHAR(128) NOT NULL WITH
DEFAULT "",
TBNAME          VARCHAR(128) NOT NULL,
TBCREATOR       VARCHAR(128) NOT NULL WITH
DEFAULT "",
COLNAMES        CLOB(2M) NOT NULL,
UNIQUERULE      CHAR(1) NOT NULL WITH DEFAULT
",
COLCOUNT       SMALLINT NOT NULL WITH DEFAULT
0,
IID             SMALLINT NOT NULL WITH DEFAULT 0,
NLEAF           INTEGER NOT NULL WITH DEFAULT 0,
NLEVELS         SMALLINT NOT NULL WITH DEFAULT 0,
FIRSTKEYCARD    BIGINT NOT NULL WITH DEFAULT
0,
FULLKEYCARD     BIGINT NOT NULL WITH DEFAULT
0,
CLUSTERRATIO    SMALLINT NOT NULL WITH
DEFAULT 0,
CLUSTERFACTOR   DOUBLE NOT NULL WITH
DEFAULT 0,
USERDEFINED     SMALLINT NOT NULL WITH
DEFAULT 0,
SYSTEM_REQUIRED SMALLINT NOT NULL WITH
DEFAULT 0,
CREATE_TIME     TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
STATS_TIME      TIMESTAMP WITH DEFAULT
CURRENT TIMESTAMP,
PAGE_FETCH_PAIRS VARCHAR(254) NOT NULL WITH
DEFAULT "",
REMARKS         VARCHAR(254) WITH DEFAULT "",
DEFINER         VARCHAR(128) NOT NULL WITH DEFAULT
",
CONVERTED       CHAR(1) NOT NULL WITH DEFAULT
",
SEQUENTIAL_PAGES INTEGER NOT NULL WITH
DEFAULT 0,
DENSITY         INTEGER NOT NULL WITH DEFAULT 0,
FIRST2KEYCARD   BIGINT NOT NULL WITH DEFAULT
0,
FIRST3KEYCARD   BIGINT NOT NULL WITH DEFAULT
0,
FIRST4KEYCARD   BIGINT NOT NULL WITH DEFAULT
0,
PCTFREE         SMALLINT NOT NULL WITH DEFAULT -
1,

```

```

    UNIQUE_COLCOUNT SMALLINT NOT NULL WITH
DEFAULT -1,
    MINPCTUSED SMALLINT NOT NULL WITH DEFAULT
0,
    REVERSE_SCANS CHAR(1) NOT NULL WITH
DEFAULT 'N',
    USE_INDEX CHAR(1),
    CREATION_TEXT CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT "",
    PACKED_DESC BLOB(1M) NOT LOGGED,
    RUN_ID TIMESTAMP,
    INDEXTYPE VARCHAR(4) NOT NULL WITH DEFAULT
",
    EXISTS CHAR(1) NOT NULL WITH DEFAULT 'N',
    RIDTOBLOCK CHAR(1) NOT NULL WITH DEFAULT
'N',
                                FOREIGN KEY (RUN_ID)
                                REFERENCES ADVISE_INSTANCE
(START_TIME)
                                ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_WORKLOAD
--
CREATE TABLE ADVISE_WORKLOAD (
    WORKLOAD_NAME CHAR(128) NOT NULL WITH
DEFAULT 'WKO',
    STATEMENT_NO INTEGER NOT NULL WITH
DEFAULT 1,
    STATEMENT_TEXT CLOB(2M) NOT NULL NOT
LOGGED,
    STATEMENT_TAG VARCHAR(256) NOT NULL WITH
DEFAULT "",
    FREQUENCY INTEGER NOT NULL WITH DEFAULT
1,
    IMPORTANCE DOUBLE NOT NULL WITH DEFAULT
1,
    WEIGHT DOUBLE NOT NULL WITH DEFAULT 1,
    COST_BEFORE DOUBLE,
    COST_AFTER DOUBLE,
    COMPILABLE CHAR(17))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_MQT
--
CREATE TABLE ADVISE_MQT (
    EXPLAIN_REQUESTER VARCHAR(128) NOT NULL WITH
DEFAULT "",
    EXPLAIN_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
    SOURCE_NAME VARCHAR(128) NOT NULL WITH
DEFAULT "",
    SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT "",
    SOURCE_VERSION VARCHAR(64) NOT NULL WITH
DEFAULT "",
    EXPLAIN_LEVEL CHAR(1) NOT NULL WITH DEFAULT
",
    STMTNO INTEGER NOT NULL WITH DEFAULT 0,
    SECTNO INTEGER NOT NULL WITH DEFAULT 0,
    NAME VARCHAR(128) NOT NULL,
    CREATOR VARCHAR(128) NOT NULL WITH
DEFAULT "",
    IID SMALLINT NOT NULL WITH DEFAULT 0,
    CREATE_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
    STATS_TIME TIMESTAMP WITH DEFAULT
CURRENT_TIMESTAMP,

```

```

    NUMROWS DOUBLE NOT NULL WITH DEFAULT
0,
    NUMCOLS SMALLINT NOT NULL WITH DEFAULT
0,
    ROWSIZE DOUBLE NOT NULL WITH DEFAULT 0,
    BENEFIT FLOAT NOT NULL WITH DEFAULT 0.0,
    USE_MQT CHAR(1),
    MQT_SOURCE CHAR(1),
    QUERY_TEXT CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT "",
    CREATION_TEXT CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT "",
    SAMPLE_TEXT CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT "",
    COLSTATS CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT "",
    EXTRA_INFO BLOB(2M) NOT NULL NOT LOGGED
with default BLOB(""),
    TBSPACE VARCHAR(128) NOT NULL WITH DEFAULT
",
    RUN_ID TIMESTAMP,
    REFRESH_TYPE CHAR(1) NOT NULL WITH
DEFAULT "",
    EXISTS CHAR(1) NOT NULL WITH DEFAULT 'N',
                                FOREIGN KEY (RUN_ID)
                                REFERENCES ADVISE_INSTANCE
(START_TIME)
                                ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_PARTITION
--
CREATE TABLE ADVISE_PARTITION (
    EXPLAIN_REQUESTER VARCHAR(128) NOT NULL WITH
DEFAULT "",
    EXPLAIN_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
    SOURCE_NAME VARCHAR(128) NOT NULL WITH
DEFAULT "",
    SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT "",
    SOURCE_VERSION VARCHAR(64) NOT NULL WITH
DEFAULT "",
    EXPLAIN_LEVEL CHAR(1) NOT NULL WITH DEFAULT
",
    STMTNO INTEGER NOT NULL WITH DEFAULT 0,
    SECTNO INTEGER NOT NULL WITH DEFAULT 0,
    QUERYNO INTEGER NOT NULL WITH DEFAULT
0,
    QUERYTAG CHAR(20) NOT NULL WITH DEFAULT
",
    TBNAME VARCHAR(128) NOT NULL,
    TBCREATOR VARCHAR(128) NOT NULL WITH
DEFAULT "",
    PMID SMALLINT NOT NULL,
    TBSPACE VARCHAR(128) NOT NULL WITH DEFAULT
",
    COLNAMES CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT "",
    COLCOUNT SMALLINT NOT NULL WITH DEFAULT
0,
    REPLICATE CHAR(1) NOT NULL WITH DEFAULT
'N',
    COST DOUBLE NOT NULL,
    USEIT CHAR(1),
    RUN_ID TIMESTAMP,
                                FOREIGN KEY(RUN_ID)
                                REFERENCES ADVISE_INSTANCE
(START_TIME)
                                ON DELETE CASCADE)

```



```

IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_TABLE
--
CREATE TABLE ADVISE_TABLE (
  RUN_ID      TIMESTAMP,
  TABLE_NAME VARCHAR(128) NOT NULL,
  TABLE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT "",
  TABLESPACE VARCHAR(128) NOT NULL WITH
DEFAULT "",
  SELECTION_FLAG VARCHAR(8) NOT NULL WITH
DEFAULT "",
  TABLE_EXISTS CHAR(1) NOT NULL WITH DEFAULT
",
  USE_TABLE CHAR(1) NOT NULL WITH DEFAULT ",
  GEN_COLUMNS CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT ",
  ORGANIZE_BY CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT ",
  CREATION_TEXT CLOB(2M) NOT NULL NOT LOGGED
WITH DEFAULT ",
  ALTER_COMMAND CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
  DISKUSE DOUBLE NOT NULL WITH DEFAULT 0 ,
  FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)

IN USERSPACE1
INDEX IN USERSPACE1;

--
-- Commit work
--
COMMIT WORK;

--
-- Optional Indexes: The following indexes are recommended for
improved performance
-- of explain-related utilities. These create index statements can be
deleted, or
-- the indexes dropped if space is a problem.
--
CREATE INDEX STMT_I1 on
  EXPLAIN_STATEMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX ARG_I1 on
  EXPLAIN_ARGUMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX PRD_I1 on
  EXPLAIN_PREDICATE(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX OPR_I1 on
  EXPLAIN_OPERATOR(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX STM_I1 on
  EXPLAIN_STREAM(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX OBJ_I1 on
  EXPLAIN_OBJECT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);

CREATE INDEX IDX_I1 on
  ADVISE_INDEX (EXPLAIN_TIME);
CREATE INDEX IDX_I2 on
  ADVISE_INDEX (NAME, EXPLAIN_TIME);
CREATE INDEX MQT_I1 on
  ADVISE_MQT (EXPLAIN_TIME);

```

```

CREATE INDEX MQT_I2 on
  ADVISE_MQT (NAME, EXPLAIN_TIME);
CREATE INDEX PRT_I1 on
  ADVISE_PARTITION (EXPLAIN_TIME);

```

```

--
-- Commit work
--
COMMIT WORK;

```

utils/UNEXPLAIN.ddl

```

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

```

```

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

```

tpccCom/comreg.h

// compreg.h : Declaration of the CCompReg

#pragma once

```

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

```

```

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

```

DECLARE_NO_REGISTRY()

```

BEGIN_COM_MAP(CCompReg)
  COM_INTERFACE_ENTRY(IComponentRegistrar)

```

```

    COM_INTERFACE_ENTRY(IDispatch)
END_COM_MAP()

// IComponentRegistrar
public:
    STDMETHOD(Attach)(BSTR bstrPath)
    {
        return S_OK;
    }
    STDMETHOD(RegisterAll)()
    {
        return _AtlComModule.RegisterServer(TRUE);
    }
    STDMETHOD(UnregisterAll)()
    {
        _AtlComModule.UnregisterServer(TRUE);
        return S_OK;
    }
    STDMETHOD(GetComponents)(SAFEARRAY **ppCLSIDs,
SAFEARRAY **ppDescriptions)
    {
        if( ppCLSIDs == NULL || ppDescriptions == NULL )
            return E_POINTER;
        int nComponents = 0;
        for (_ATL_OBJMAP_ENTRY** ppEntry =
_AtlComModule.m_ppAutoObjMapFirst; ppEntry <
_AtlComModule.m_ppAutoObjMapLast; ppEntry++)
        {
            if (*ppEntry != NULL)
            {
                _ATL_OBJMAP_ENTRY* pEntry = *ppEntry;
                if (pEntry->pclsid != NULL)
                {
                    LPCTSTR pszDescription = pEntry-
>pfnGetObjectDescription();
                    if (pszDescription)
                        nComponents++;
                }
            }
        }
        SAFEARRAYBOUND rgBound[1];
        rgBound[0].lbound = 0;
        rgBound[0].cElements = nComponents;
        *ppCLSIDs = SafeArrayCreate(VT_BSTR, 1, rgBound);
        if( *ppCLSIDs == NULL )
            return AtlHresultFromLastError();
        *ppDescriptions = SafeArrayCreate(VT_BSTR, 1,
rgBound);
        if( *ppDescriptions == NULL )
            return AtlHresultFromLastError();
        LONG i = 0;
        for (_ATL_OBJMAP_ENTRY** ppEntry =
_AtlComModule.m_ppAutoObjMapFirst; ppEntry <
_AtlComModule.m_ppAutoObjMapLast; ppEntry++)
        {
            if (*ppEntry != NULL)
            {
                _ATL_OBJMAP_ENTRY* pEntry = *ppEntry;
                if (pEntry->pclsid != NULL)
                {
                    LPCTSTR pszDescription = pEntry-
>pfnGetObjectDescription();
                    if (pszDescription)
                    {
                        LPCTSTR 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 PrxDllGetClassObject(REFCLSID rclsid, REFIID riid,
LPVOID* ppv);
    STDAPI PrxDllRegisterServer(void);
    STDAPI PrxDllUnregisterServer(void);
}

#endif

```

tpccCom/Resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.

```

```

// Used by tpccCom.rc
//
#define IDS_PROJNAME            100
#define IDR_TPCCCOM            101
#define IDR_TPCC_COM           102

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        201
#define _APS_NEXT_COMMAND_VALUE        32768
#define _APS_NEXT_CONTROL_VALUE        201
#define _APS_NEXT_SYMED_VALUE          103
#endif
#endif

tpccCom/stdafx.h

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

#pragma once

#ifndef STRICT
#define STRICT
#endif

// Modify the following defines if you have to target a platform prior to
// the ones specified below.
// Refer to MSDN for the latest info on corresponding values for
// different platforms.
#ifndef WINVER                // Allow use of features specific
// to Windows 95 and Windows NT 4 or later.
#define WINVER 0x0400        // Change this to the appropriate
// value to target Windows 98 and Windows 2000 or later.
#endif

#ifndef _WIN32_WINNT          // Allow use of features specific to
// Windows NT 4 or later.
#define _WIN32_WINNT 0x0400 // Change this to the appropriate
// value to target Windows 2000 or later.
#endif

#ifndef _WIN32_WINDOWS       // Allow use of features specific
// to Windows 98 or later.
#define _WIN32_WINDOWS 0x0410 // Change this to the
// appropriate value to target Windows Me or later.
#endif

#ifndef _WIN32_IE            // Allow use of features specific to IE
// 4.0 or later.
#define _WIN32_IE 0x0400    // Change this to the appropriate
// value to target IE 5.0 or later.
#endif

#define _ATL_APARTMENT_THREADED
#define _ATL_NO_AUTOMATIC_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 <atlcocom.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 "..\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)(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:
    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/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 "compreg.h"
#include "dlldatax.h"
```

```
class CtpccComModule : public CAtlDllModuleT< CtpccComModule
>
{
public :
    DECLARE_LIBID(LIBID_tpccComLib)
    DECLARE_REGISTRY_APPID_RESOURCEID(IDR_TPCCCO
M, "{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_paym.s_W_ID<<" d_id:"<< pymt-
>in_paym.s_D_ID<<"
           " s_transtatus:"<<pymt-
>out_paym.s_transtatus<<endl);

    do_pymt(pymt,connectHandle);

    DEBUGMSG("Return from do_pymt call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<pymt->in_paym.s_W_ID<<" d_id:"<< pymt-
>in_paym.s_D_ID<<"
           " s_transtatus:"<<pymt-
>out_paym.s_transtatus<<endl);

    DEBUGMSG("Connection handle set to free" <<endl);
    connectHandleInUse = 0;

    return S_OK;
}

/*
*****
** Name      :    doOrderStatus
** Description :
**           Call db2 dll entry point to execute txn
** Parameters :
**           int*      size of UCHAR buffer to pay
attention to
**           UCHAR**   char buffer that holds txn
wrapper struct
** Returns   :
**           int - return code
** Comments  :
**
*****

```

```

**
*****
*/

STDMETHODIMP Ctpcc_com::doOrderStatus(INT* size, UCHAR**
buffer)
{
    ords_wrapper *ords;
    ords = (ords_wrapper *) *buffer;

    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to
true"<<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use."<<endl);
        ERRORMSG("Context handle in use."<<endl);
        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_ords call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<ords->in_ords.s_W_ID<<" d_id:"<< ords-
>in_ords.s_D_ID<<"
           " s_transtatus:"<<ords-
>out_ords.s_transtatus<<endl);

    do_ords(ords,connectHandle);

    DEBUGMSG("Return from do_ords call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<"
w_id:"<<ords->in_ords.s_W_ID<<" d_id:"<< ords-
>in_ords.s_D_ID<<"
           " s_transtatus:"<<ords-
>out_ords.s_transtatus<<endl);

    DEBUGMSG("Connection handle set to free" <<endl);
    connectHandleInUse = 0;

    return S_OK;
}

/*
*****
** Name      :    doDBInfo
** Description :
**           Function to test com interface
** Parameters :
** Returns   :
**           int - return code
** Comments  :
**
*****
*/

STDMETHODIMP Ctpcc_com::doDBInfo(void)
{
    DEBUGMSG("Stub function to warm object pool"<<endl);
    return S_OK;
}

/*
*****
** Name      :    loadLibrary
** Description :
**           Function loads appropiate db library based
on
**           registry setting
*****

```

```

** Parameters :
** Returns    :
**           : int - return code
** Comments   :
*****
*/

Ctpcc_com::loadLibrary()
{
    DEBUGMSG("Entered loadLibrary function"<<endl);
    //check to see if dbInstance is already loaded
    if(!dbInstance)
    {
        DEBUGMSG("Database dll not loaded. Loading
dll."<<endl);
        if (nullDB)
        {
            DEBUGMSG("Loading "<<dbType << " nulldb dll." <<
endl);
            dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\nullDB.dll");
            if(dbInstance == NULL)
            {
                DEBUGMSG("Unable to load null db dll,
rc:"<<GetLastError());
                ERRORMSG("Unable to load null db dll,
rc:"<<GetLastError());
                return ERR_NULL_DLL_NOT_LOADED;
            }
            DEBUGMSG(dbType << " nulldb dll loaded"<<endl);
        }
        else if(strcmp(dbType,"DB2") == 0)
        {
            DEBUGMSG("Loading "<<dbType << " dll." << endl);

            dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\tpccDB2glue.dll");
            if(dbInstance == NULL)
            {
                DEBUGMSG("Unable to load library."<<endl);

                ERRORMSG("Unable to load com dll, rc:" <<
GetLastError() << endl);

                return ERR_DB2_DLL_NOT_LOADED;
            }
            DEBUGMSG(dbType<< " dll loaded"<<endl);
        }
        else if( strcmp(dbType,"ORACLE") == 0 )
        {
            DEBUGMSG("Unable to load oracle dll"<<endl);
            ERRORMSG("Unable to load oracle dll,
rc:"<<GetLastError()<<endl);
            return ERR_ORACLE_DLL_NOT_LOADED;
        }
        else
        {
            DEBUGMSG("Unknown database type
dll:"<<dbType<<endl);
            ERRORMSG("Unknown database type
dll:"<<dbType<<endl);
            return ERR_UNKNOWN_DB;
        }

        //retrieve function addresses from instance loaded.
        DEBUGMSG("Getting do_connection function address
from "<<dbType<<" dll"<<endl);
        if( do_connection =
(CONNECT_PTR)GetProcAddress(dbInstance,"connect_db") ==
NULL )
            return ERR_CONNECT_ADDRESS_NOT_FOUND;

```

```

        DEBUGMSG("do_connection
address:"<<DEBUGADDRESS(do_connection)<<endl);

        DEBUGMSG("Getting do_disconnect function address
from "<<dbType<<" dll"<<endl);
        if( (do_disconnect =
(DISCONNECT_PTR)GetProcAddress(dbInstance,"disconnect_db")
) == NULL )
            return
ERR_DISCONNECT_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_disconnect
address:"<<DEBUGADDRESS(do_disconnect)<<endl);

        DEBUGMSG("Getting do_nord function address from
"<<dbType<<" dll"<<endl);
        if( (do_nord = (NORD_PTR)
GetProcAddress(dbInstance,"do_nord") == NULL)
            return ERR_NORD_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_nord function
address:"<<DEBUGADDRESS(do_nord)<<endl);

        DEBUGMSG("Getting do_pymt function address from
"<<dbType<<" dll"<<endl);
        if( (do_pymt = (PYMT_PTR)
GetProcAddress(dbInstance,"do_pymt") == NULL)
            return ERR_PYMT_ADDRESS_NOT_FOUND;
        DEBUGMSG("do_pymt function
address:"<<DEBUGADDRESS(do_pymt)<<endl);

        DEBUGMSG("Getting do_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_SU
B_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,(BYT
E *)&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,0x44,0x
00,0xaa,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,0
x2D,0x6C,0xE3,0xE7,0x52);
```

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

```
MIDL_DEFINE_GUID(CLSID,
CLSID_tpcc_com,0x5F752BF2,0xF739,0x43D4,0x84,0x92,0x44,0x
C1,0x95,0x81,0xC0,0xA1);
```

```
#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,0x10,0
x48,0x60}},{2,0}}};

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

```

```

extern const MIDL_SERVER_INFO
IComponentRegistrar_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
IComponentRegistrar_ProxyInfo;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO Itpcc_com_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
Itpcc_com_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(__RPC_WIN32__)
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT50_OR_LATER)
#error You need a Windows 2000 or later to run this stub because it
uses these features:
#error /robust command line switch.
#error However, your C/C++ compilation flags indicate you intend to
run this app on earlier systems.
#error This app will 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 GetComponent */
/* 96 */ 0x33, /* FC_AUTO_HANDLE */
          0x6c, /* Old Flags: object, Oi2 */
/* 98 */ NdrFcLong( 0x0 ), /* 0 */
/* 102 */ NdrFcShort( 0xa ), /* 10 */
/* 104 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 106 */ NdrFcShort( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x8 ), /* 8 */
/* 110 */ 0x45, /* Oi2 Flags: srv must size, has return, has
ext, */
          0x3, /* 3 */
/* 112 */ 0x8, /* 8 */
          0x3, /* Ext Flags: new corr desc, clt corr check,
*/
/* 114 */ NdrFcShort( 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( 0xffd2 ), /* 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( 0xffff ), /* -8 */
/* 288 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 290 */ NdrFcShort( 0x2 ), /* Offset= 2 (292) */
/* 292 */ NdrFcShort( 0x10 ), /* 16 */
/* 294 */ NdrFcShort( 0x2f ), /* 47 */
/* 296 */ NdrFcLong( 0x14 ), /* 20 */
/* 300 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/*
*/
/* 302 */ NdrFcLong( 0x3 ), /* 3 */
/* 306 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 308 */ NdrFcLong( 0x11 ), /* 17 */
/* 312 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 314 */ NdrFcLong( 0x2 ), /* 2 */
/* 318 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/*
*/
/* 320 */ NdrFcLong( 0x4 ), /* 4 */

```

```

/* 324 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT
*/
/* 326 */ NdrFcLong( 0x5 ), /* 5 */
/* 330 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 332 */ NdrFcLong( 0xb ), /* 11 */
/* 336 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT
*/
/* 338 */ NdrFcLong( 0xa ), /* 10 */
/* 342 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 344 */ NdrFcLong( 0x6 ), /* 6 */
/* 348 */ NdrFcShort( 0xe8 ), /* Offset= 232 (580) */
/* 350 */ NdrFcLong( 0x7 ), /* 7 */
/* 354 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 356 */ NdrFcLong( 0x8 ), /* 8 */
/* 360 */ NdrFcShort( 0xe2 ), /* Offset= 226 (586) */
/* 362 */ NdrFcLong( 0xd ), /* 13 */
/* 366 */ NdrFcShort( 0xff3c ), /* Offset= -196 (170) */
/* 368 */ NdrFcLong( 0x9 ), /* 9 */
/* 372 */ NdrFcShort( 0xff6e ), /* Offset= -146 (226) */
/* 374 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 378 */ NdrFcShort( 0xd4 ), /* Offset= 212 (590) */
/* 380 */ NdrFcLong( 0x24 ), /* 36 */
/* 384 */ NdrFcShort( 0xd6 ), /* Offset= 214 (598) */
/* 386 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 390 */ NdrFcShort( 0xd0 ), /* Offset= 208 (598) */
/* 392 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 396 */ NdrFcShort( 0x100 ), /* Offset= 256 (652) */
/* 398 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 402 */ NdrFcShort( 0xfe ), /* Offset= 254 (656) */
/* 404 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 408 */ NdrFcShort( 0xfc ), /* Offset= 252 (660) */
/* 410 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 414 */ NdrFcShort( 0xfa ), /* Offset= 250 (664) */
/* 416 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 420 */ NdrFcShort( 0xf8 ), /* Offset= 248 (668) */
/* 422 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 426 */ NdrFcShort( 0xf6 ), /* Offset= 246 (672) */
/* 428 */ NdrFcLong( 0x400b ), /* 16395 */
/* 432 */ NdrFcShort( 0xe0 ), /* Offset= 224 (656) */
/* 434 */ NdrFcLong( 0x400a ), /* 16394 */
/* 438 */ NdrFcShort( 0xde ), /* Offset= 222 (660) */
/* 440 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 444 */ NdrFcShort( 0xe8 ), /* Offset= 232 (676) */
/* 446 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 450 */ NdrFcShort( 0xde ), /* Offset= 222 (672) */
/* 452 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 456 */ NdrFcShort( 0xe0 ), /* Offset= 224 (680) */
/* 458 */ NdrFcLong( 0x400d ), /* 16397 */
/* 462 */ NdrFcShort( 0xde ), /* Offset= 222 (684) */
/* 464 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 468 */ NdrFcShort( 0xdc ), /* Offset= 220 (688) */
/* 470 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 474 */ NdrFcShort( 0xda ), /* Offset= 218 (692) */
/* 476 */ NdrFcLong( 0x400c ), /* 16396 */
/* 480 */ NdrFcShort( 0xe0 ), /* Offset= 224 (704) */
/* 482 */ NdrFcLong( 0x10 ), /* 16 */
/* 486 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 488 */ NdrFcLong( 0x12 ), /* 18 */
/* 492 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT
*/
/* 494 */ NdrFcLong( 0x13 ), /* 19 */
/* 498 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 500 */ NdrFcLong( 0x15 ), /* 21 */
/* 504 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER
*/
/* 506 */ NdrFcLong( 0x16 ), /* 22 */
/* 510 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 512 */ NdrFcLong( 0x17 ), /* 23 */
/* 516 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 518 */ NdrFcLong( 0xe ), /* 14 */

```

```

/* 522 */ NdrFcShort( 0xbe ), /* Offset= 190 (712) */
/* 524 */ NdrFcLong( 0x400e ), /* 16398 */
/* 528 */ NdrFcShort( 0xc2 ), /* Offset= 194 (722) */
/* 530 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 534 */ NdrFcShort( 0xc0 ), /* Offset= 192 (726) */
/* 536 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 540 */ NdrFcShort( 0x74 ), /* Offset= 116 (656) */
/* 542 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 546 */ NdrFcShort( 0x72 ), /* Offset= 114 (660) */
/* 548 */ NdrFcLong( 0x4015 ), /* 16405 */
/* 552 */ NdrFcShort( 0x70 ), /* Offset= 112 (664) */
/* 554 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 558 */ NdrFcShort( 0x66 ), /* Offset= 102 (660) */
/* 560 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 564 */ NdrFcShort( 0x60 ), /* Offset= 96 (660) */
/* 566 */ NdrFcLong( 0x0 ), /* 0 */
/* 570 */ NdrFcShort( 0x0 ), /* Offset= 0 (570) */
/* 572 */ NdrFcLong( 0x1 ), /* 1 */
/* 576 */ NdrFcShort( 0x0 ), /* Offset= 0 (576) */
/* 578 */ NdrFcShort( 0xffff ), /* Offset= -1 (577) */
/* 580 */
0x15, /* FC_STRUCT */
0x7, /* 7 */
/* 582 */ NdrFcShort( 0x8 ), /* 8 */
/* 584 */ 0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 586 */
0x13, 0x0, /* FC_OP */
/* 588 */ NdrFcShort( 0xfdc6 ), /* Offset= -570 (18) */
/* 590 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 592 */ NdrFcShort( 0x2 ), /* Offset= 2 (594) */
/* 594 */
0x13, 0x0, /* FC_OP */
/* 596 */ NdrFcShort( 0x1b8 ), /* Offset= 440 (1036) */
/* 598 */
0x13, 0x0, /* FC_OP */
/* 600 */ NdrFcShort( 0x20 ), /* Offset= 32 (632) */
/* 602 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 604 */ NdrFcLong( 0x2f ), /* 47 */
/* 608 */ NdrFcShort( 0x0 ), /* 0 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 614 */ 0x0, /* 0 */
0x0, /* 0 */
/* 616 */ 0x0, /* 0 */
0x0, /* 0 */
/* 618 */ 0x0, /* 0 */
0x46, /* 70 */
/* 620 */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 622 */ NdrFcShort( 0x1 ), /* 1 */
/* 624 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 626 */ NdrFcShort( 0x4 ), /* 4 */
/* 628 */ NdrFcShort( 0x1 ), /* Corr flags: early */
/* 630 */ 0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 632 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 634 */ NdrFcShort( 0x10 ), /* 16 */
/* 636 */ NdrFcShort( 0x0 ), /* 0 */
/* 638 */ NdrFcShort( 0xa ), /* Offset= 10 (648) */
/* 640 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 642 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */

```

```

/* 644 */ NdrFcShort( 0xffd6 ), /* Offset= -42 (602) */
/* 646 */ 0x36, /* FC_POINTER */
/* 648 */ 0x5b, /* FC_END */
/* 650 */ 0x13, 0x0, /* FC_OP */
/* 652 */ NdrFcShort( 0xffe2 ), /* Offset= -30 (620) */
/* 654 */ 0x13, 0x8, /* FC_OP [simple_pointer] */
/* 656 */ 0x1, /* FC_BYTE */
/* 658 */ 0x5c, /* FC_PAD */
/* 660 */ 0x13, 0x8, /* FC_OP [simple_pointer] */
/* 662 */ 0x8, /* FC_LONG */
/* 664 */ 0x5c, /* FC_PAD */
/* 666 */ 0x13, 0x8, /* FC_OP [simple_pointer] */
/* 668 */ 0xb, /* FC_HYPER */
/* 670 */ 0x5c, /* FC_PAD */
/* 672 */ 0x13, 0x8, /* FC_OP [simple_pointer] */
/* 674 */ 0xa, /* FC_FLOAT */
/* 676 */ 0x5c, /* FC_PAD */
/* 678 */ 0x13, 0x8, /* FC_OP [simple_pointer] */
/* 680 */ 0xc, /* FC_DOUBLE */
/* 682 */ 0x5c, /* FC_PAD */
/* 684 */ 0x13, 0x0, /* FC_OP */
/* 686 */ NdrFcShort( 0xff9e ), /* Offset= -98 (580) */
/* 688 */ 0x13, 0x10, /* FC_OP [pointer_deref] */
/* 690 */ NdrFcShort( 0xffa0 ), /* Offset= -96 (586) */
/* 692 */ 0x13, 0x10, /* FC_OP [pointer_deref] */
/* 694 */ NdrFcShort( 0xfdfc ), /* Offset= -516 (170) */
/* 696 */ 0x13, 0x10, /* FC_OP [pointer_deref] */
/* 698 */ NdrFcShort( 0xfe30 ), /* Offset= -464 (226) */
/* 700 */ 0x13, 0x10, /* FC_OP [pointer_deref] */
/* 702 */ NdrFcShort( 0x2 ), /* Offset= 2 (696) */
/* 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 */
/* 714 */ 0x7, /* 7 */
/* 716 */ NdrFcShort( 0x10 ), /* 16 */
/* 718 */ 0x6, /* FC_SHORT */
/* 720 */ 0x1, /* FC_BYTE */
/* 722 */ 0x1, /* FC_BYTE */
/* 724 */ 0x8, /* FC_LONG */
/* 726 */ 0xb, /* FC_HYPER */
/* 728 */ 0x5b, /* FC_END */
/* 730 */ 0x13, 0x0, /* FC_OP */
/* 732 */ NdrFcShort( 0xffff4 ), /* Offset= -12 (712) */
/* 734 */ 0x13, 0x8, /* FC_OP [simple_pointer] */
/* 736 */ 0x2, /* FC_CHAR */
/* 738 */ 0x5c, /* FC_PAD */
/* 740 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 742 */ 0x7, /* 7 */
/* 744 */ NdrFcShort( 0x20 ), /* 32 */
/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ NdrFcShort( 0x0 ), /* Offset= 0 (736) */
/* 750 */ 0x8, /* FC_LONG */
/* 752 */ 0x8, /* FC_LONG */
/* 754 */ 0x6, /* FC_SHORT */
/* 756 */ 0x6, /* FC_SHORT */
/* 758 */ 0x6, /* FC_SHORT */
/* 760 */ 0x6, /* FC_SHORT */
/* 762 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 764 */ 0x0, /* 0 */
/* 766 */ NdrFcShort( 0xfe30 ), /* Offset= -464 (282) */
/* 768 */ 0x5c, /* FC_PAD */
/* 770 */ 0x5b, /* FC_END */
/* 772 */ 0x1b, /* FC_CARRAY */
/* 774 */ 0x3, /* 3 */
/* 776 */ NdrFcShort( 0x4 ), /* 4 */
/* 778 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 780 */ 0x0, /* 0 */
/* 782 */ NdrFcShort( 0x0 ), /* 0 */
/* 784 */ NdrFcShort( 0x1 ), /* Corr flags: early */
/* 786 */ 0x4b, /* FC_PP */
/* 788 */ 0x5c, /* FC_PAD */
/* 790 */ 0x48, /* FC_VARIABLE_REPEAT */
/* 792 */ 0x49, /* FC_FIXED_OFFSET */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x0 ), /* 0 */
/* 798 */ NdrFcShort( 0x1 ), /* 1 */
/* 800 */ NdrFcShort( 0x0 ), /* 0 */
/* 802 */ NdrFcShort( 0x0 ), /* 0 */
/* 804 */ 0x13, 0x0, /* FC_OP */
/* 806 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (730) */
/* 808 */ 0x5b, /* FC_END */
/* 810 */ 0x8, /* FC_LONG */
/* 812 */ 0x5c, /* FC_PAD */
/* 814 */ 0x5b, /* FC_END */
/* 816 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 818 */ 0x3, /* 3 */
/* 820 */ NdrFcShort( 0x8 ), /* 8 */
/* 822 */ NdrFcShort( 0x0 ), /* 0 */
/* 824 */ NdrFcShort( 0x6 ), /* Offset= 6 (794) */
/* 826 */ 0x8, /* FC_LONG */
/* 828 */ 0x36, /* FC_POINTER */
/* 830 */ 0x5c, /* FC_PAD */
/* 832 */ 0x5b, /* FC_END */
/* 834 */ 0x11, 0x0, /* FC_RP */
/* 836 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (750) */
/* 838 */ 0x1b, /* FC_CARRAY */
/* 840 */ 0x3, /* 3 */
/* 842 */ NdrFcShort( 0x4 ), /* 4 */
/* 844 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 846 */ 0x0, /* 0 */
/* 848 */ NdrFcShort( 0x0 ), /* 0 */
/* 850 */ NdrFcShort( 0x1 ), /* Corr flags: early */
/* 852 */ 0x4b, /* FC_PP */
/* 854 */ 0x5c, /* FC_PAD */
/* 856 */ 0x48, /* FC_VARIABLE_REPEAT */

```

```

0x49, /* FC_FIXED_OFFSET */
/* 812 */ NdrFcShort( 0x4 ), /* 4 */
/* 814 */ NdrFcShort( 0x0 ), /* 0 */
/* 816 */ NdrFcShort( 0x1 ), /* 1 */
/* 818 */ NdrFcShort( 0x0 ), /* 0 */
/* 820 */ NdrFcShort( 0x0 ), /* 0 */
/* 822 */ 0x13, 0x0, /* FC_OP */
/* 824 */ NdrFcShort( 0xff40 ), /* Offset= -192 (632) */
/* 826 */
0x5b, /* FC_END */

0x8, /* FC_LONG */
/* 828 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 830 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 832 */ NdrFcShort( 0x8 ), /* 8 */
/* 834 */ NdrFcShort( 0x0 ), /* 0 */
/* 836 */ NdrFcShort( 0x6 ), /* Offset= 6 (842) */
/* 838 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 840 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 842 */
0x11, 0x0, /* FC_RP */
/* 844 */ NdrFcShort( 0xffd2 ), /* Offset= -46 (798) */
/* 846 */
0x1d, /* FC_SMFARRAY */
0x0, /* 0 */
/* 848 */ NdrFcShort( 0x8 ), /* 8 */
/* 850 */ 0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 852 */
0x15, /* FC_STRUCT */
0x3, /* 3 */
/* 854 */ NdrFcShort( 0x10 ), /* 16 */
/* 856 */ 0x8, /* FC_LONG */
0x6, /* FC_SHORT */
/* 858 */ 0x6, /* FC_SHORT */
0x4c, /* FC_EMBEDDED_COMPLEX */
/* 860 */ 0x0, /* 0 */
NdrFcShort( 0xffff1 ), /* Offset= -15 (846) */
0x5b, /* FC_END */
/* 864 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 866 */ NdrFcShort( 0x18 ), /* 24 */
/* 868 */ NdrFcShort( 0x0 ), /* 0 */
/* 870 */ NdrFcShort( 0xa ), /* Offset= 10 (880) */
/* 872 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 874 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 876 */ NdrFcShort( 0xffe8 ), /* Offset= -24 (852) */
/* 878 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 880 */
0x11, 0x0, /* FC_RP */
/* 882 */ NdrFcShort( 0xfd4a ), /* Offset= -694 (188) */
/* 884 */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 886 */ NdrFcShort( 0x1 ), /* 1 */
/* 888 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* 0 */
/* 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, /* 0 */
/* 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, /* 0 */
/* 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,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: IDispatch, ver. 0.0,
GUID={0x00020400,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: IComponentRegistrar, ver. 0.0,
GUID={0xa817e7a2,0x43fa,0x11d0,{0x9e,0x44,0x00,0xaa,0x00,0xb6,0x77,0x0a}} */

#pragma code_seg(".orpc")

```

```

static const unsigned short
IComponentRegistrar_FormatStringOffsetTable[] =
{
    (unsigned short) -1,
    (unsigned short) -1,
    (unsigned short) -1,
    (unsigned short) -1,
    0,
    36,
    66,
    96,
    138,
    174
};

static const MIDL_STUBLESS_PROXY_INFO
IComponentRegistrar_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &IComponentRegistrar_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

static const MIDL_SERVER_INFO IComponentRegistrar_ServerInfo
=
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &IComponentRegistrar_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0};

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

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

```

```

NdrStubCall2,
NdrStubCall2,
NdrStubCall2
};

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

/* Object interface: Itpcc_com, ver. 0.0,

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

#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};

INTERFACE_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,

```

```

    &Itppcc_com_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x6000169, /* MIDL Version 6.0.361 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

const CInterfaceProxyVtbl * _tpccCom_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &Itppcc_comProxyVtbl,
    ( CInterfaceProxyVtbl *) &IComponentRegistrarProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpccCom_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &Itppcc_comStubVtbl,
    ( CInterfaceStubVtbl *) &IComponentRegistrarStubVtbl,
    0
};

PCInterfaceName const _tpccCom_InterfaceNamesList[] =
{
    "Itppcc_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 */
};
#endif
#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

```

```

#ifdef 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
*paym, void *ctx);
extern "C" TPCCDB2GLUE_API int do_orcs(ords_wrapper
*ords, void *ctx);
extern "C" TPCCDB2GLUE_API int do_dlvly(dlvly_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::i
n | 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_SU
B_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, (BY
TE *) &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, &regTy
pe, (BYTE *) &value, &regValueSize) == ERROR_SUCCESS )
                    strcpy(userPassword, value);
                else

```



```

        return ERR_INVALID_PASSWORD;
        DEBUGMSG("DB user password:"<<userPassword
<< endl);
    }
    else
    {
        return ERR_INVALID_REGISTRY_KEY;
        DEBUGMSG("Unable to open registry key"<<
REGISTRY_SUB_KEY << endl);
    }

    break;
case DLL_THREAD_ATTACH:
    break;
case DLL_THREAD_DETACH:
    break;
case DLL_PROCESS_DETACH:
    #ifdef TIMING
        ERRORMSG("dll_process_detach called, closing
timing file"<<endl);
        fclose(respTimes);
    #endif
    break;
}
return TRUE;
}

/*
*****
** Name      :    attachContext
** Description :
**           Function calls db2 api to attach thread to
**           a specific context per thread basis.
** Parameters :
**           void*    stored context
** Returns   :
**           int - return code
** Comments  :
**
*****
*/
extern "C" int attachContext(void *ctx)
{
    int rc;
    if ( (rc = attach_context(ctx)) != OK)
        return ERR_ATTACHING_CONTEXT;

    return OK;
}

/*
*****
** Name      :    detachContext
** Description :
**           Function calls db2 api to detach thread
**           from context
** Parameters :
**           void*    stored context
** Returns   :
**           int - return code
** Comments  :
**
*****
*/
extern "C" int detachContext(void *ctx)
{
    int rc;
    if ( (rc = detach_context(ctx)) != OK)
    {
        ERRORMSG("error detaching context from db,
rc:"<<rc<<endl);
        return ERR_DETACHING_CONTEXT;

```

```

    }
    return OK;
}

/*
*****
** Name      :    connect_db
** Description :
**           Function calls db2 api to connect to db
** Parameters :
**           char*    dbName
**           void**   uninitialized context
** Returns   :
**           int - return code
** Comments  :
**           To connect to db, first connection must be
**           established. Next, context for that connect
**           be saved off. Finally, detach from the
**           context just created.
**
*****
*/
extern "C" TPCCDB2GLUE_API int connect_db(char *dbName,void
**ctx)
{
    DEBUGMSG("Entered db2glue do_connect using dbName:"<<
dbName << endl << "Calling connect_to_TM_auth() with
username:"<< userName << " password:" <<userPassword <<
endl);

    int rc =
connect_to_TM_auth(dbName,userName,userPassword);
    if(rc != OK)
    {
        DEBUGMSG("Object do_connect failed, rc:"<<rc<<endl);
        ERRORMSG("Object do_connect failed, rc:"<<rc<<endl);

        return rc;
    }

    DEBUGMSG("calling get_context"<<endl);
    if ( (rc = get_context(ctx)) != OK)
    {
        DEBUGMSG("Object get_context() failed, rc:"<< rc
<<endl);
        ERRORMSG("Object get_context() failed, rc:"<< rc
<<endl);

        return ERR_SAVING_CONTEXT;
    }
    DEBUGMSG("Object get_context successful, context:"<<
DEBUGADDRESS(*ctx)<<" saved"<<endl);

    DEBUGMSG("Object calling detach_context() w/
ctx:"<<DEBUGADDRESS(*ctx)<<"<<endl);
    if( (rc = detach_context(*ctx)) != OK)
    {
        DEBUGMSG("Object failed detach_context w/
ctx:"<<DEBUGADDRESS(*ctx)<<" rc:" << rc << endl);
        ERRORMSG("Object failed detach_context w/
ctx:"<<DEBUGADDRESS(*ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("Object detach_context successful,
context:"<<DEBUGADDRESS(*ctx)<<" , connection
complete"<<endl);

    return OK;
}

/*

```

```

*****
** Name      : disconnect_db
** Description :
**           Function calls db2 api to disconnect from
db
** Parameters :
**           void*      stored context
** Returns   :
**           int - return code
** Comments  :

**           To disconnect from db, first must attach to
**           thread's context. Next, disconnect from db
*****
*/

extern "C" TPCCDB2GLUE_API int disconnect_db(void *ctx)
{
    DEBUGMSG("Entered do_disconnect, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("context established. preparing to call db2" <<
endl);

    rc = disconnect_from_TM();
    if(rc != OK)
    {
        DEBUGMSG("disconnect failed, rc:"<<rc<<endl);
        ERRORMSG("disconnect failed, rc:"<<rc<<endl);
        return rc;
    }
    return OK;
}

/*
*****
** Name      : do_nord
** Description :
**           Function calls db2 api to execute nord txn
** Parameters :
**           nord_wrapper* new order txn structs
wrapper
**           void*      stored context
** Returns   :
**           int - return code
** Comments  :
**           Attach to thread's context, call nord sql
function
**           then detach from context.
*****
*/

extern "C" TPCCDB2GLUE_API int do_nord(nord_wrapper
*nord,void *ctx)
{
    DEBUGMSG("Entered do_nord, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("nord failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("nord failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
    }
}

```

```

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to context:" <<
DEBUGADDRESS(ctx)<<" preparing to call db2" << endl);

#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif

    //call new order txn
    neword_sql(&nord->in_nord,&nord->out_nord);

#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=1;
    EnterCriticalSection(&errorMutex);
    rc = fwrite(&timeSample,sizeof(struct txn),1,respTimes);

    LeaveCriticalSection(&errorMutex);
#endif
    DEBUGMSG("return from neword_sql(), s_transtatus:" <<
nord->out_nord.s_transtatus << endl);

    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("nord failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("nord failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }

    return OK;
}

/*
*****
** Name      : do_pymt
** Description :
**           Function calls db2 api to execute pymt txn
** Parameters :
**           paym_wrapper* payment txn structs
wrapper
**           void*      stored context
** Returns   :
**           int - return code
** Comments  :
**           Attach to thread's context, call nord sql
function
**           then detach from context.
*****
*/

extern "C" TPCCDB2GLUE_API int do_pymt(paym_wrapper
*pymt,void *ctx)
{
    DEBUGMSG("Entered do_pymt, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("pymt failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("pymt failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
    }
}

```

```

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to context:" << DEBUGADDRESS(ctx)
<<" preparing to call db2" << endl);

#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif

    //call pymt txn
    payment_sql(&pymt->in_paym,&pymt->out_paym);

#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=2;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
    {
        ERRORMSG("Unable to write to binary file, pymt"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif

    DEBUGMSG("return from payment_sql(), s_transtatus:" <<
pymt->out_paym.s_transtatus << endl);

    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("pymt failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<endl);
        DEBUGMSG("pymt failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("pymt detach_context successful. pymt txn
complete."<<endl);

    return OK;
}

/*
*****
** Name      :    do_ords
** Description :    Function calls db2 api to execute ords txn
** Parameters :    ords_wrapper* order status txn structs
wrapper
**           :    void*          stored context
** Returns   :
**           :    int - return code
** Comments  :
**           :    Attach to thread's context, call nord sql
function
**           :    then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_ords(ords_wrapper
*ords,void *ctx)
{
    DEBUGMSG("Entered do_ords, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("ords failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

```

```

        DEBUGMSG("ords failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to
context:"<<DEBUGADDRESS(ctx)<<" , preparing to call db2" <<
endl);

    DEBUGMSG("calling ordstat_sql()" <<endl);

#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
    ordstat_sql(&ords->in_ords,&ords->out_ords);
#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=3;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
    {
        ERRORMSG("Unable to write to binary file, ords"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif

    DEBUGMSG("return from ordstat_sql(), s_transtatus:" << ords-
>out_ords.s_transtatus << endl);

    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("ords failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("ords failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("ords detach_context successful. pymt txn
complete."<<endl);

    return OK;
}

/*
*****
** Name      :    do_dlvly
** Description :    Function calls db2 api to execute ords txn
** Parameters :    dlvly_wrapper* dlvly txn structs wrapper
**           :    void*          stored context
** Returns   :
**           :    int - return code
** Comments  :
**           :    Attach to thread's context, call nord sql
function
**           :    then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_dlvly(dlvly_wrapper *dlvly,void
*ctx)
{
    DEBUGMSG("Entered do_dlvly, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("dlvly failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

```

```

        DEBUGMSG("dlvy failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to
context:"<<DEBUGADDRESS(ctx)<<" , preparing to call db2" <<
endl);

    DEBUGMSG("calling delivery_sql" << endl);
#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif

    //call dlvy txn
    delivery_sql(&dlvy->in_dlvy,&dlvy->out_dlvy);
#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=3;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
    {
        ERRORMSG("Unable to write to binary file, dlvy"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif

    DEBUGMSG("return from delivery_sql(), s_transtatus:" << dlvy-
>out_dlvy.s_transtatus << endl);

    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("dlvy failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("dlvy failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }

    DEBUGMSG("dlvy detach_context successful. dlvy txn
complete."<<endl);

    return OK;
}

/*
*****
** Name      :    do_stok
** Description :
**           Function calls db2 api to execute stok txn
** Parameters :
**           stok_wrapper* stok txn structs wrapper
**           void*        stored context
** Returns   :
**           int - return code
** Comments  :
**           Attach to thread's context, call nord sql
function
**           then detach from context.
*****
*/

extern "C" TPCCDB2GLUE_API int do_stok(stok_wrapper
*stok,void *ctx)
{
    DEBUGMSG("Entered do_stok, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)

```

```

    {
        ERRORMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attaching to
context:"<<DEBUGADDRESS(ctx)<<" , preparing to call db2" <<
endl);

    DEBUGMSG("calling stocklev_sql()" <<endl);

#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif

    //call stock level txn
    stocklev_sql(&stok->in_stok, &stok->out_stok);

#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=5;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1 )
    {
        ERRORMSG("Unable to write to binary file, stok"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif

    DEBUGMSG("return from stocklev_sql(), s_transtatus:" <<
stok->out_stok.s_transtatus << endl);

    DEBUGMSG("calling detach_context"<<endl);
    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("detach_context successful. stok txn
complete."<<endl);

    return OK;
}

```

Appendix - B: :Tunable Parameters

B.1 Database Parameters.

db.cfg.out

Database Configuration for Database TPCC

Database configuration release level = 0x0a00
 Database release level = 0x0a00

Database territory = US
 Database code page = 819
 Database code set = ISO8859-1
 Database country/region code = 1
 Database collating sequence = BINARY
 Alternate collating sequence (ALT_COLLATE) =

Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database (DISCOVER_DB) = ENABLE

Default query optimization class (DFT_QUERYOPT) = 5
 Degree of parallelism (DFT_DEGREE) = 1
 Continue upon arithmetic exceptions (DFT_SQLMATHWARN) = NO
 Default refresh age (DFT_REFRESH_AGE) = 0
 Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
 Number of frequent values retained (NUM_FREQVALUES) = 10
 Number of quantiles retained (NUM_QUANTILES) = 20

Backup pending = NO

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

Multi-page file allocation enabled = YES

Log retain for recovery status = RECOVERY
 User exit for logging status = NO

Data Links Token Expiry Interval (sec) (DL_EXPINT) = 60
 Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60
 Data Links Number of Copies (DL_NUM_COPIES) = 1
 Data Links Time after Drop (days) (DL_TIME_DROP) = 1
 Data Links Token in Uppercase (DL_UPPER) = NO
 Data Links Token Algorithm (DL_TOKEN) = MAC0

Database heap (4KB) (DBHEAP) = 300000
 Size of database shared memory (4KB) (DATABASE_MEMORY) = 95400000
 Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*4)
 Log buffer size (4KB) (LOGBUFSZ) = 60000
 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) = 10000

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) = 20
 Number of I/O servers (NUM_IOSERVERS) = 1
 Index sort flag (INDEXSORT) = YES
 Sequential detect flag (SEQDETECT) = NO
 Default prefetch size (pages) (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages (TRACKMOD) = OFF

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

Max number of active applications (MAXAPPLS) = 1000
 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/rdbfasttloglv
 Overflow log path (OVERFLOWLOGPATH) =
 Mirror log path (MIRRORLOGPATH) =
 First active log file = S0000001.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) = 5075
 Log retain for recovery enabled (LOGRETAIN) = RECOVERY
 User exit for logging enabled (USEREXIT) = OFF

HADR database role = STANDARD
 HADR local host name (HADR_LOCAL_HOST) =
 HADR local service name (HADR_LOCAL_SVC) =
 HADR remote host name (HADR_REMOTE_HOST) =
 HADR remote service name (HADR_REMOTE_SVC) =
 HADR instance name of remote server (HADR_REMOTE_INST) =
 HADR timeout value (HADR_TIMEOUT) = 120
 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC

First log archive method (LOGARCHMETH1) = LOGRETAIN
 Options for logarchmeth1 (LOGARCHOPT1) =
 Second log archive method (LOGARCHMETH2) = OFF
 Options for logarchmeth2 (LOGARCHOPT2) =
 Failover log archive path (FAILARCHPATH) =
 Number of log archive retries on error (NUMARCHRETRY) = 5
 Log archive retry Delay (secs) (ARCHRETRYDELAY) = 20
 Vendor options (VENDOROPT) =

Auto restart enabled (AUTORESTART) = ON
Index re-creation time and redo index build (INDEXREC) =
SYSTEM (RESTART)
Log pages during index build (LOGINDEXBUILD) = OFF
Default number of loadrec sessions (DFT_LOADREC_SES) = 1
Number of database backups to retain (NUM_DB_BACKUPS) =
12
Recovery history retention (days) (REC_HIS_RETENTN) = 366

TSM management class (TSM_MGMTCLASS) =
TSM node name (TSM_NODENAME) =
TSM owner (TSM_OWNER) =
TSM password (TSM_PASSWORD) =

Automatic maintenance (AUTO_MAINT) = OFF
Automatic database backup (AUTO_DB_BACKUP) = OFF
Automatic table maintenance (AUTO_TBL_MAINT) = OFF
Automatic runstats (AUTO_RUNSTATS) = OFF
Automatic statistics profiling (AUTO_STATS_PROF) = OFF
Automatic profile updates (AUTO_PROF_UPD) = OFF
Automatic reorganization

dbm.cfg.out

Database Manager Configuration

Node type = Database Server with local clients

Database manager configuration release level = 0x0a00

CPU speed (millisec/instruction) (CPUSPEED) =
3.778755e-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) = 3
Diagnostic data directory path (DIAGPATH) =

Default database monitor switches

Buffer pool (DFT_MON_BUFPOOL) = OFF
Lock (DFT_MON_LOCK) = OFF
Sort (DFT_MON_SORT) = OFF
Statement (DFT_MON_STMT) = OFF
Table (DFT_MON_TABLE) = OFF
Timestamp (DFT_MON_TIMESTAMP) = OFF
Unit of work (DFT_MON_UOW) = OFF
Monitor health of instance and databases (HEALTH_MON) = OFF

SYSADM group name (SYSADM_GROUP) = STAFF
SYSCTRL group name (SYSCTRL_GROUP) =
SYSMAINT group name (SYSMAINT_GROUP) =
SYSMON group name (SYSMON_GROUP) =

Client Userid-Password Plugin (CLNT_PW_PLUGIN) =
Client Kerberos Plugin (CLNT_KRB_PLUGIN) =
Group Plugin (GROUP_PLUGIN) =
GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) =
Server Plugin Mode (SRV_PLUGIN_MODE) =
UNFENCED
Server List of GSS Plugins (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin (SRVCON_PW_PLUGIN) =

Server Connection Authentication (SRVCON_AUTH) =
NOT_SPECIFIED
Database manager authentication (AUTHENTICATION) =
CLIENT
Cataloging allowed without authority (CATALOG_NOAUTH) =
YES
Trust all clients (TRUST_ALLCLNTS) = YES
Trusted client authentication (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication (FED_NOAUTH) = NO

Default database path (DFTDBPATH) = /home/tpcc

Database monitor heap size (4KB) (MON_HEAP_SZ) = 4096
Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) = 1024
Audit buffer size (4KB) (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB) (INSTANCE_MEMORY) =
AUTOMATIC
Backup buffer default size (4KB) (BACKBUFSZ) = 1024
Restore buffer default size (4KB) (RESTBUFSZ) = 1024

Sort heap threshold (4KB) (SHEAPTHRES) = 20000

Directory cache support (DIR_CACHE) = YES

Application support layer heap size (4KB) (ASLHEAPSZ) = 15
Max requester I/O block size (bytes) (RQRIOBLK) = 4096
Query heap size (4KB) (QUERY_HEAP_SZ) = 1000

Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 10

Priority of agents (AGENTPRI) = 60
Max number of existing agents (MAXAGENTS) = 1000
Agent pool size (NUM_POOLAGENTS) = 700
Initial number of agents in pool (NUM_INITAGENTS) = 0
Max number of coordinating agents (MAX_COORDAGENTS) =
MAXAGENTS
Max no. of concurrent coordinating agents (MAXCAGENTS) =
MAX_COORDAGENTS
Max number of client connections (MAX_CONNECTIONS) =
MAX_COORDAGENTS

Keep fenced process (KEEPFENCED) = YES
Number of pooled fenced processes (FENCED_POOL) =
MAX_COORDAGENTS
Initial number of fenced processes (NUM_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) =
RESTART

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

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

TCP/IP Service name (SVCENAME) = db2ctppc
Discovery mode (DISCOVER) = SEARCH
Discover server instance (DISCOVER_INST) = ENABLE

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

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) =
512
Number of FCM request blocks (FCM_NUM_RQB) =
AUTOMATIC
Number of FCM connection entries (FCM_NUM_CONNECT) =
AUTOMATIC

Number of FCM message anchors (FCM_NUM_ANCHORS) =
AUTOMATIC

affinity.cfg

```
<RESOURCE_POLICY>  
<GLOBAL_RESOURCE_POLICY>  
<METHOD>RSET</METHOD>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00000</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00001</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00002</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00003</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00004</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00005</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00006</RESOURCE>  
</RESOURCE_BINDING>  
<RESOURCE_BINDING>  
<RESOURCE>sys/node.03.00007</RESOURCE>  
</RESOURCE_BINDING>  
</GLOBAL_RESOURCE_POLICY>  
</RESOURCE_POLICY>
```

db2set.cfg.out

```
[i] DB2_RESOURCE_POLICY=/home/tpcc/tpc-c.ibm/cfg/affinity.cfg  
[i] DB2_SELUDI_COMM_BUFFER=Y  
[i] DB2_USE_ALTERNATE_PAGE_CLEANING=YES  
[i] DB2_MAX_NON_TABLE_LOCKS=500  
[i] DB2_LGPGPAGE_BP=YES  
[i] DB2_TRUSTED_BINDIN=ON  
[i] DB2_KEEPTABLELOCK=ON  
[i] DB2_NO_FORK_CHECK=ON  
[i] DB2_APM_PERFORMANCE=ALL  
[i] DB2_ENABLE_BUFDP=OFF  
[i] DB2_PINNED_BP=YES  
[i] DB2_SELECTIVITY=ON  
[i] DB2ASSUMEUPDATE=ON  
[i] DB2CHECKCLIENTINTERVAL=0  
[i] DB2_HASH_JOIN=OFF  
[i] DB2CHKSQLDA=OFF  
[i] DB2ENVLIST=MEMORY_AFFINITY_LDR_CNTRL  
[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 24
Maximum pool size 24
Creation timeout 100,000
Enable Object Construction

Enable Just in time activation
Concurrency Required

InetInfo registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\In  
etInfo\Parameters]  
"ListenBackLog"=dword:000000fa  
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,0  
0,00,00,00,00  
"MaxConnections"=dword:000061a8  
"PoolThreadLimit"=dword:00000190  
"ThreadTimeout"=dword:00015180  
"MaxConcurrency"=dword:ffffff
```

tcPIP parameters registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tc  
pip\Parameters]  
"NV Hostname"="client32"  
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,  
52,00,6f,00,6f,\
```

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

```
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00  
,00,00
```

```
"NameServer"=""  
"ForwardBroadcasts"=dword:00000000  
"IPEnableRouter"=dword:00000000
```

```
"Domain"=""  
"Hostname"="client32"  
"SearchList"=""
```

```
"UseDomainNameDevolution"=dword:00000001  
"EnableICMPRedirect"=dword:00000001  
"DeadGWDetectDefault"=dword:00000001  
"DontAddDefaultGatewayDefault"=dword:00000000  
"EnableSecurityFilters"=dword:00000000  
"AllowUnqualifiedQuery"=dword:00000000  
"PrioritizeRecordData"=dword:00000001  
"GlobalTcpWindowSize"=dword:00008000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tc  
pip\Parameters\Adapters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tc  
pip\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,00,36,00,31,00,44,00,33,00,33  
,00,39,00,\
```

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

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

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

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

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

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

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

00,42,00,36,00,30,00,34,00,39,00,36,00,36,00,44,00,43,00,7d,00,00,00,00,00

"NumInterfaces"=dword:00000002

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

46,58,7e,40,82,50,f0,3b,60,49,66,dc

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{0435C97F-9186-473F-B181-5449A2CF0042}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,00,34,00,33,00,35,00,43,00,39,00,\

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

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

46,00,30,00,30,00,34,00,32,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{1E07A95A-92A0-4836-BF73-7AE38F8ACA07}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,00,45,00,30,00,37,00,41,00,39,00,\

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

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

41,00,43,00,41,00,30,00,37,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{2EA04AA5-93A6-437F-9153-2F6834D3B795}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

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

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

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

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

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{37430121-7BE3-4B55-8AAB-D8AD09B2029C}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,33,00,37,00,34,00,33,00,30,00,31,00,\

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

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

32,00,30,00,32,00,39,00,43,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{6FE29D81-59D5-4401-A77E-BE3BC929B6E0}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,00,46,00,45,00,32,00,39,00,44,00,\

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

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

39,00,42,00,36,00,45,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{7B215199-A3F3-4836-89A6-390C5E70E801}]

"LLInterface"=""

"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

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

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

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

30,00,45,00,38,00,30,00,31,00,7d,00,00,00,00,00


```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{A32BB4A3-C9B2-4ADB-A65D-18BB314BF7F0}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,33,00,32,00,42,00,42,00,34,00,\

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

00,41,00,36,00,35,00,44,00,2d,00,31,00,38,00,42,00,42,00,33,00,31,00,34,00,\
42,00,46,00,37,00,46,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{A71EB7B5-37C6-42DB-BE8F-BB231FD1BE00}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,37,00,31,00,45,00,42,00,37,00,\

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

00,42,00,45,00,38,00,46,00,2d,00,42,00,42,00,32,00,33,00,31,00,46,00,44,00,\
31,00,42,00,45,00,30,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{BEABCC14-9C0A-4BE9-9817-14C4092418D3}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,42,00,45,00,41,00,42,00,43,00,43,00,\

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

00,39,00,38,00,31,00,37,00,2d,00,31,00,34,00,43,00,34,00,30,00,39,00,32,00,\
34,00,31,00,38,00,44,00,33,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{CD3F7746-9E60-4E22-9A40-7BC6CC6B2E2E}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

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

00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,00,44,00,33,00,46,00,37,00,37,00,\
```

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

00,39,00,41,00,34,00,30,00,2d,00,37,00,42,00,43,00,36,00,43,00,43,00,36,00,\
42,00,32,00,45,00,32,00,45,00,7d,00,00,00,00,00

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

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

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{0435C97F-9186-473F-B181-5449A2CF0042}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,33,00,35,00,2e,00,31,00,2e,00,31,00,2e,00,31,00,00,00,\
00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{1E07A95A-92A0-4836-BF73-7AE38F8ACA07}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,\
00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,\
0,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{2EA04AA5-93A6-437F-9153-2F6834D3B795}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,36,00,38,00,2e,00,31,00,31,00,\
2e,00,35,00,31,00,00,00,00,00
```



```

"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A71EB7B5-37C6-42DB-BE8F-BB231FD1BE00}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,33,00,32,00,2e,00,31,00,2e,00,32,00,00,00,\
00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,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,\
0,30,00,30,00,\
32,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:40be0640
"T1"=dword:40be0d48
"T2"=dword:40be128e
"LeaseTerminatesTime"=dword:40be1450
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{BEABCC14-9C0A-4BE9-9817-14C4092418D3}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,32,00,2e,00,31,00,00,00,00,00,\
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):00,00
"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{CD3F7746-9E60-4E22-9A40-7BC6CC6B2E2E}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,\
00,00,00

```

```

"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,0,0,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\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,\
00,00,00,01,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,00,00,06,00,00,00,00,00,00,06,00,00,00,00,00,06,00,00,00,02,00,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,00,00,02,00,00,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00,00,00,00

```

Tpcc software registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"divyLogPath"="c:\inetpub\wwwroot\tpcc\divy"
"divyQueueLen"=dword:00004e20
>nullDB"=dword:00000000
"htmlTrace"=dword:00000000
"dbName"="tpcc"
"errorLogFile"="c:\inetpub\wwwroot\tpcc\isapi_err.log"
"htmlTraceLogFile"="c:\inetpub\wwwroot\tpcc\isapi.log"
"numUsers"=dword:00005208
"dbType"="DB2"
"dbUserName"="tpcc"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\inetpub\wwwroot\tpcc\db2glue.dll"
"divyThreads"=dword:00000005

```



```

dedicated true Partition is dedicated
False
ent_capacity 1600 Entitled processor capacity
False
frequency 0 System Bus Frequency
False
fullcore false Enable full CORE dump
True
fwversion IBM,SF220_001 Firmware version and revision
levels False
id_to_partition 0X8000000250600001 Partition ID
False
id_to_system 0X8000000250600000 System ID
False
iostat false Continuously maintain DISK I/O history
True
keylock normal State of system keylock at boot time
False
max_capacity 1600 Maximum potential processor
capacity False
max_logname 9 Maximum login name length at boot
time True
maxbuf 20 Maximum number of pages in block I/O
BUFFER CACHE True
maxmbuf 0 Maximum Kbytes of real memory
allowed for MBUFS True
maxpout 0 HIGH water mark for pending write I/Os
per file True
maxuproc 20000 Maximum number of PROCESSES
allowed per user True
min_capacity 100 Minimum potential processor capacity
False
minpout 0 LOW water mark for pending write I/Os
per file True
modelname IBM,9117-570 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 527695872 Amount of usable physical memory
in Kbytes False
rtasversion 1 Open Firmware RTAS version
False
systemid IBM,0210007BA Hardware system identifier
False
variable_weight 0 Variable processor capacity weight
False

```

vmo -L

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT
TYPE						

DEPENDENCIES

```

-----
memory_frames 125M 125M 4KB pages
S
-----
pinnable_frames 10554K 10554K 4KB pages
S
-----
maxfree 128 128 128 16 200K 4KB pages
D
  minfree
  memory_frames
-----
minfree 120 120 120 8 200K 4KB pages
D
  maxfree
  memory_frames
-----

```

```

minperm% 20 20 20 1 100 % memory
D
  maxperm%
-----
minperm 2151K 2151K S
-----
maxperm% 80 80 80 1 100 % memory
D
  minperm%
  maxclient%
-----
maxperm 8604K 8604K S
-----
strict_maxperm 0 0 0 0 1 boolean D
-----
maxpin% 99 80 99 1 99 % memory D
  pinnable_frames
  memory_frames
-----
maxpin 128679K 128679K S
-----
maxclient% 80 80 80 1 100 % memory D
  maxperm%
-----
lrubucket 128K 128K 128K 64K 4KB pages
D
-----
defps 1 1 1 0 1 boolean D
-----
nokilluid 0 0 0 0 4G-1 uid D
-----
numpsbkls 512K 512K 4KB pages S
-----
npskill 4K 4K 4K 1 512K-1 4KB pages D
-----
npswam 16K 16K 16K 0 512K-1 4KB pages
D
-----
v_pinshm 1 0 1 0 1 boolean D
-----
pta_balance_threshold n/a 1 1 0 99 % pta segment
D
-----
pagecoloring n/a 0 0 0 1 boolean B
-----
framesets 2 2 2 1 10 B
-----
mempools 1 1 1 1 32 B
-----
lgpg_size 16M 0 16M 0 16M bytes D
  lgpg_regions
-----
lgpg_regions 28401 0 28700 0 D
  lgpg_size
-----
num_spec_dataseg 0 0 0 0 B
-----
spec_dataseg_int 512 512 512 0 B
-----
memory_affinity 1 1 1 0 1 boolean B
-----
htabscale n/a -1 -1 -4 0 B
-----
force_relalias_lite 0 0 0 0 1 boolean D
-----
relalias_percentage 0 0 0 0 32K-1 D
-----
rpgcontrol 0 0 0 0 3 D
-----
rpgclean 0 0 0 0 1 boolean D
-----

```

npsrpgmin	24K	24K	24K	0	512K-1	4KB	pages	
D								
npsrpgmax								

npsrpgmax	32K	32K	32K	0	512K-1	4KB	pages	
D								
npsrpgmin								

scrub	0	0	0	0	1	boolean		D

scrubclean	0	2	2	0	1	boolean		D

npsscrubmin	24K	24K	24K	0	512K-1	4KB	pages	
D								
npsscrubmax								

npsscrubmax	32K	32K	32K	0	512K-1	4KB	pages	
D								
npsscrubmin								

data_stagger_interval	161	161	161	0	4K-1	pages		
D								
lgpg_regions								

large_page_heap_size	0	0	0	0	8E-1			B
lgpg_regions								

kernel_heap_psize	4K	4K	4K	4K	16M			B
lgpg_regions								

soft_min_lgpgs_vmpool	0	0	0	0	90			D
lgpg_regions								

vm_modlist_threshold	-1	-1	-1	-2	2G-1			D

vmm_fork_policy	1	1	1	0	1	boolean		D

low_ps_handling	1	1	1	1	2			D

n/a means parameter not supported by the current platform or kernel

Parameter types:

- S = Static: cannot be changed
- D = Dynamic: can be freely changed
- B = Bosboot: can only be changed using bosboot and reboot
- R = Reboot: can only be changed during reboot
- C = Connect: changes are only effective for future socket connections
- M = Mount: changes are only effective for future mountings
- I = Incremental: can only be incremented

Value conventions:

- K = Kilo: 2¹⁰
- G = Giga: 2³⁰
- P = Peta: 2⁵⁰
- M = Mega: 2²⁰
- T = Tera: 2⁴⁰
- E = Exa: 2⁶⁰

ioo -L

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
DEPENDENCIES							

minpgahead	2	2	2	0	4K	4KB pages	D
maxpgahead							

maxpgahead	8	8	8	0	4K	4KB pages	D
minpgahead							

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

maxrandwrt	0	0	0	0	512K	4KB pages	D

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

numfsbufs	196	196	196	1	2G-1			M

sync_release_ilock	0	0	0	0	1	boolean		D

lvm_bufcnt	9	9	9	1	64	128KB/buffer		D

j2_minPageReadAhead	2	2	2	0	64K	4KB pages		D

j2_maxPageReadAhead	8	8	8	0	64K	4KB pages		D

j2_nBufferPerPagerDevice	512	512	512	0	2G-1			M

j2_nPagesPerWriteBehindCluster	32	32	32	0	64K			D

j2_maxRandomWrite	0	0	0	0	64K	4KB pages		D

j2_nRandomCluster	0	0	0	0	64K	16KB clusters		D

jfs_cread_enabled	0	0	0	0	1	boolean		D

jfs_use_read_lock	1	1	1	0	1	boolean		D

hd_pvs_opn	0	0						S

j2_inodeCacheSize	400	400	400	1	1000			D

j2_metadataCacheSize	400	400	400	1	1000			D

pv_min_pbuf	0	512	512	512	2G-1			D

j2_dynamicBufferPreallocation	16	16	16	0	256	16k slabs		M

j2_maxUsableMaxTransfer	512	512	512	1	4K	pages		M

n/a means parameter not supported by the current platform or kernel

Parameter types:

- S = Static: cannot be changed
- D = Dynamic: can be freely changed
- B = Bosboot: can only be changed using bosboot and reboot
- R = Reboot: can only be changed during reboot
- C = Connect: changes are only effective for future socket connections
- M = Mount: changes are only effective for future mountings
- I = Incremental: can only be incremented

Value conventions:

- K = Kilo: 2¹⁰
- G = Giga: 2³⁰
- P = Peta: 2⁵⁰
- M = Mega: 2²⁰
- T = Tera: 2⁴⁰
- E = Exa: 2⁶⁰

Appendix - C: Database Setup Code

C.1 Database Creation Scripts

db/create_database.ddl

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

ts/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/rD1F01V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F01V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F01V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F01V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
```

```
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/rD1F02V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F02V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F02V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F02V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F03V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F03V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F03V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F03V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F04V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F04V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F04V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F04V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F05V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F05V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F05V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F05V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F06V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F06V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F06V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F06V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F07V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F07V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F07V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F07V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F08V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F08V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F08V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
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/rD1F08V4CST' 10081536
    )
    extentsize 256
    prefetchsize 4096;
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/rD1F09V1CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
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/rD1F09V2CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
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/rD1F09V3CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
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/rD1F09V4CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
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/rD1F10V1CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
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/rD1F10V2CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_customer_39 of D1

drop tablespace ts_customer_39;
create regular tablespace ts_customer_39 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V3CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_customer_40 of D1

drop tablespace ts_customer_40;
create regular tablespace ts_customer_40 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V4CST' 10081536
)
    extentsize 256
    prefetchsize 4096;
commit;

connect reset;

connect to tpcc;
-- 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/rD1F01V1CSTI' 319872
)
    extentsize 256
    prefetchsize 4096
    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/rD1F01V2CSTI' 319872
)
    extentsize 256
    prefetchsize 4096
    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/rD1F01V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F01V4CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F02V1CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F02V2CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F02V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F02V4CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F03V1CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F03V2CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F03V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F03V4CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F04V1CSTI' 319872
    )
    extentsize 256
    prefetchsize 4096
    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/rD1F04V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F04V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F04V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F05V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F05V2CSTI' 319872

```

```

    )
    extentsize 256
    prefetchsize 4096
    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/rD1F05V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F05V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F06V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F06V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F06V3CSTI' 319872
)
extentsize 256

```

```

        prefetchsize 4096
        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/rD1F06V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F07V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F07V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F07V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F07V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F08V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F08V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F08V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F08V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F09V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
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/rD1F09V2CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F09V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F09V4CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F10V1CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F10V2CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_39 of D1

```

```

drop tablespace is_customer_39;
create regular tablespace is_customer_39 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_40 of D1

drop tablespace is_customer_40;
create regular tablespace is_customer_40 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V4CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  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' 1024,
    device '/dev/rD1F01V2DIST' 1024,
    device '/dev/rD1F01V3DIST' 1024,
    device '/dev/rD1F01V4DIST' 1024,
    device '/dev/rD1F02V1DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
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/rD1F02V2DIST' 1024,
    device '/dev/rD1F02V3DIST' 1024,
    device '/dev/rD1F02V4DIST' 1024,
    device '/dev/rD1F03V1DIST' 1024,
    device '/dev/rD1F03V2DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
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/rD1F03V3DIST' 1024,
    device '/dev/rD1F03V4DIST' 1024,
    device '/dev/rD1F04V1DIST' 1024,
    device '/dev/rD1F04V2DIST' 1024,
    device '/dev/rD1F04V3DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
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/rD1F04V4DIST' 1024,
    device '/dev/rD1F05V1DIST' 1024,
    device '/dev/rD1F05V2DIST' 1024,
    device '/dev/rD1F05V3DIST' 1024,
    device '/dev/rD1F05V4DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
commit;

```

-- now creating TS for ts_dist_05 of D1

```

drop tablespace ts_dist_05;
create regular tablespace ts_dist_05 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V1DIST' 1024,
    device '/dev/rD1F06V2DIST' 1024,
    device '/dev/rD1F06V3DIST' 1024,
    device '/dev/rD1F06V4DIST' 1024,
    device '/dev/rD1F07V1DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
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/rD1F07V2DIST' 1024,
    device '/dev/rD1F07V3DIST' 1024,
    device '/dev/rD1F07V4DIST' 1024,
    device '/dev/rD1F08V1DIST' 1024,
    device '/dev/rD1F08V2DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
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/rD1F08V3DIST' 1024,
    device '/dev/rD1F08V4DIST' 1024,
    device '/dev/rD1F09V1DIST' 1024,
    device '/dev/rD1F09V2DIST' 1024,
    device '/dev/rD1F09V3DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
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/rD1F09V4DIST' 1024,
    device '/dev/rD1F10V1DIST' 1024,
    device '/dev/rD1F10V2DIST' 1024,
    device '/dev/rD1F10V3DIST' 1024,
    device '/dev/rD1F10V4DIST' 1024
  )
  extentsize 64
  prefetchsize 4096;
commit;

```

connect reset;

ts/crts_hist.ddl

connect to tpcc;
-- 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' 284416,
    device '/dev/rD1F01V2HIST' 284416,
    device '/dev/rD1F01V3HIST' 284416,
    device '/dev/rD1F01V4HIST' 284416,
    device '/dev/rD1F02V1HIST' 284416
  )
  extentsize 256
  prefetchsize 4096
  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/rD1F02V2HIST' 284416,
    device '/dev/rD1F02V3HIST' 284416,
    device '/dev/rD1F02V4HIST' 284416,
    device '/dev/rD1F03V1HIST' 284416,
    device '/dev/rD1F03V2HIST' 284416
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

```

-- now creating TS for ts_history_03 of D1

```

drop tablespace ts_history_03;
create regular tablespace ts_history_03 pagesize 16K

```

```

managed by database
using
(
    device '/dev/rD1F03V3HIST' 284416,
    device '/dev/rD1F03V4HIST' 284416,
    device '/dev/rD1F04V1HIST' 284416,
    device '/dev/rD1F04V2HIST' 284416,
    device '/dev/rD1F04V3HIST' 284416
)
extentsize 256
prefetchsize 4096
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/rD1F04V4HIST' 284416,
    device '/dev/rD1F05V1HIST' 284416,
    device '/dev/rD1F05V2HIST' 284416,
    device '/dev/rD1F05V3HIST' 284416,
    device '/dev/rD1F05V4HIST' 284416
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_05 of D1

drop tablespace ts_history_05;
create regular tablespace ts_history_05 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V1HIST' 284416,
    device '/dev/rD1F06V2HIST' 284416,
    device '/dev/rD1F06V3HIST' 284416,
    device '/dev/rD1F06V4HIST' 284416,
    device '/dev/rD1F07V1HIST' 284416
)
extentsize 256
prefetchsize 4096
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/rD1F07V2HIST' 284416,
    device '/dev/rD1F07V3HIST' 284416,
    device '/dev/rD1F07V4HIST' 284416,
    device '/dev/rD1F08V1HIST' 284416,
    device '/dev/rD1F08V2HIST' 284416
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_07 of D1

```

```

drop tablespace ts_history_07;
create regular tablespace ts_history_07 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V3HIST' 284416,
    device '/dev/rD1F08V4HIST' 284416,
    device '/dev/rD1F09V1HIST' 284416,
    device '/dev/rD1F09V2HIST' 284416,
    device '/dev/rD1F09V3HIST' 284416
)
extentsize 256
prefetchsize 4096
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/rD1F09V4HIST' 284416,
    device '/dev/rD1F10V1HIST' 284416,
    device '/dev/rD1F10V2HIST' 284416,
    device '/dev/rD1F10V3HIST' 284416,
    device '/dev/rD1F10V4HIST' 284416
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

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/rD1F01V4ITEM' 64,
    device '/dev/rD1F02V1ITEM' 64,
    device '/dev/rD1F02V2ITEM' 64,
    device '/dev/rD1F02V3ITEM' 64,
    device '/dev/rD1F02V4ITEM' 64,
    device '/dev/rD1F03V1ITEM' 64,
    device '/dev/rD1F03V2ITEM' 64,
    device '/dev/rD1F03V3ITEM' 64,
    device '/dev/rD1F03V4ITEM' 64,
    device '/dev/rD1F04V1ITEM' 64,
    device '/dev/rD1F04V2ITEM' 64,
    device '/dev/rD1F04V3ITEM' 64,
    device '/dev/rD1F04V4ITEM' 64,
    device '/dev/rD1F05V1ITEM' 64,
    device '/dev/rD1F05V2ITEM' 64,
    device '/dev/rD1F05V3ITEM' 64,
    device '/dev/rD1F05V4ITEM' 64,

```



```

device '/dev/rD1F06V1ITEM' 64,
device '/dev/rD1F06V2ITEM' 64,
device '/dev/rD1F06V3ITEM' 64,
device '/dev/rD1F06V4ITEM' 64,
device '/dev/rD1F07V1ITEM' 64,
device '/dev/rD1F07V2ITEM' 64,
device '/dev/rD1F07V3ITEM' 64,
device '/dev/rD1F07V4ITEM' 64,
device '/dev/rD1F08V1ITEM' 64,
device '/dev/rD1F08V2ITEM' 64,
device '/dev/rD1F08V3ITEM' 64,
device '/dev/rD1F08V4ITEM' 64,
device '/dev/rD1F09V1ITEM' 64,
device '/dev/rD1F09V2ITEM' 64,
device '/dev/rD1F09V3ITEM' 64,
device '/dev/rD1F09V4ITEM' 64,
device '/dev/rD1F10V1ITEM' 64,
device '/dev/rD1F10V2ITEM' 64,
device '/dev/rD1F10V3ITEM' 64,
device '/dev/rD1F10V4ITEM' 64
)
extentsize 16
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

ts/crts_neword.ddl

connect to tpcc;
-- now creating TS for ts_neworda_01 of D1

drop tablespace ts_neworda_01;
create regular tablespace ts_neworda_01 pagesize 4K
managed by database
using
(
device '/dev/rD1F01V1NORA' 148992,
device '/dev/rD1F01V2NORA' 148992,
device '/dev/rD1F01V3NORA' 148992,
device '/dev/rD1F01V4NORA' 148992,
device '/dev/rD1F02V1NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_02 of D1

drop tablespace ts_neworda_02;
create regular tablespace ts_neworda_02 pagesize 4K
managed by database
using
(
device '/dev/rD1F02V2NORA' 148992,
device '/dev/rD1F02V3NORA' 148992,
device '/dev/rD1F02V4NORA' 148992,
device '/dev/rD1F03V1NORA' 148992,
device '/dev/rD1F03V2NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_03 of D1

drop tablespace ts_neworda_03;
create regular tablespace ts_neworda_03 pagesize 4K
managed by database

```

```

using
(
device '/dev/rD1F03V3NORA' 148992,
device '/dev/rD1F03V4NORA' 148992,
device '/dev/rD1F04V1NORA' 148992,
device '/dev/rD1F04V2NORA' 148992,
device '/dev/rD1F04V3NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_04 of D1

drop tablespace ts_neworda_04;
create regular tablespace ts_neworda_04 pagesize 4K
managed by database
using
(
device '/dev/rD1F04V4NORA' 148992,
device '/dev/rD1F05V1NORA' 148992,
device '/dev/rD1F05V2NORA' 148992,
device '/dev/rD1F05V3NORA' 148992,
device '/dev/rD1F05V4NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_05 of D1

drop tablespace ts_neworda_05;
create regular tablespace ts_neworda_05 pagesize 4K
managed by database
using
(
device '/dev/rD1F06V1NORA' 148992,
device '/dev/rD1F06V2NORA' 148992,
device '/dev/rD1F06V3NORA' 148992,
device '/dev/rD1F06V4NORA' 148992,
device '/dev/rD1F07V1NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_06 of D1

drop tablespace ts_neworda_06;
create regular tablespace ts_neworda_06 pagesize 4K
managed by database
using
(
device '/dev/rD1F07V2NORA' 148992,
device '/dev/rD1F07V3NORA' 148992,
device '/dev/rD1F07V4NORA' 148992,
device '/dev/rD1F08V1NORA' 148992,
device '/dev/rD1F08V2NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_07 of D1

drop tablespace ts_neworda_07;
create regular tablespace ts_neworda_07 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V3NORA' 148992,
device '/dev/rD1F08V4NORA' 148992,

```

```

        device '/dev/rD1F09V1NORA' 148992,
        device '/dev/rD1F09V2NORA' 148992,
        device '/dev/rD1F09V3NORA' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_neword_a_08 of D1

drop tablespace ts_neword_a_08;
create regular tablespace ts_neword_a_08 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F09V4NORA' 148992,
        device '/dev/rD1F10V1NORA' 148992,
        device '/dev/rD1F10V2NORA' 148992,
        device '/dev/rD1F10V3NORA' 148992,
        device '/dev/rD1F10V4NORA' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

connect reset;

connect to tpcc;
-- now creating TS for ts_newordb_01 of D1

drop tablespace ts_newordb_01;
create regular tablespace ts_newordb_01 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V1NORB' 148992,
        device '/dev/rD1F01V2NORB' 148992,
        device '/dev/rD1F01V3NORB' 148992,
        device '/dev/rD1F01V4NORB' 148992,
        device '/dev/rD1F02V1NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_02 of D1

drop tablespace ts_newordb_02;
create regular tablespace ts_newordb_02 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F02V2NORB' 148992,
        device '/dev/rD1F02V3NORB' 148992,
        device '/dev/rD1F02V4NORB' 148992,
        device '/dev/rD1F03V1NORB' 148992,
        device '/dev/rD1F03V2NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_03 of D1

drop tablespace ts_newordb_03;
create regular tablespace ts_newordb_03 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F03V3NORB' 148992,
        device '/dev/rD1F03V4NORB' 148992,

```

```

        device '/dev/rD1F04V1NORB' 148992,
        device '/dev/rD1F04V2NORB' 148992,
        device '/dev/rD1F04V3NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_04 of D1

drop tablespace ts_newordb_04;
create regular tablespace ts_newordb_04 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F04V4NORB' 148992,
        device '/dev/rD1F05V1NORB' 148992,
        device '/dev/rD1F05V2NORB' 148992,
        device '/dev/rD1F05V3NORB' 148992,
        device '/dev/rD1F05V4NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_05 of D1

drop tablespace ts_newordb_05;
create regular tablespace ts_newordb_05 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F06V1NORB' 148992,
        device '/dev/rD1F06V2NORB' 148992,
        device '/dev/rD1F06V3NORB' 148992,
        device '/dev/rD1F06V4NORB' 148992,
        device '/dev/rD1F07V1NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_06 of D1

drop tablespace ts_newordb_06;
create regular tablespace ts_newordb_06 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F07V2NORB' 148992,
        device '/dev/rD1F07V3NORB' 148992,
        device '/dev/rD1F07V4NORB' 148992,
        device '/dev/rD1F08V1NORB' 148992,
        device '/dev/rD1F08V2NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_07 of D1

drop tablespace ts_newordb_07;
create regular tablespace ts_newordb_07 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F08V3NORB' 148992,
        device '/dev/rD1F08V4NORB' 148992,
        device '/dev/rD1F09V1NORB' 148992,
        device '/dev/rD1F09V2NORB' 148992,
        device '/dev/rD1F09V3NORB' 148992
    )

```

```

    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_08 of D1

drop tablespace ts_newordb_08;
create regular tablespace ts_newordb_08 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F09V4NORB' 148992,
        device '/dev/rD1F10V1NORB' 148992,
        device '/dev/rD1F10V2NORB' 148992,
        device '/dev/rD1F10V3NORB' 148992,
        device '/dev/rD1F10V4NORB' 148992
    )
    extentsize 256
    prefetchsize 4096;
commit;

connect reset;

```

ts/crts_order.ddl

```

connect to tpcc;
-- 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' 291584
    )
    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' 291584
    )
    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' 291584
    )
    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/rD1F01V4ORD' 291584
    )
    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/rD1F02V1ORD' 291584
    )
    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/rD1F02V2ORD' 291584
    )
    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/rD1F02V3ORD' 291584
    )
    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/rD1F02V4ORD' 291584
    )
    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/rD1F03V1ORD' 291584
    )
    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/rD1F03V2ORD' 291584
    )
    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/rD1F03V3ORD' 291584
    )
    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/rD1F03V4ORD' 291584
    )
    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/rD1F04V1ORD' 291584
    )
    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/rD1F04V2ORD' 291584
    )
    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/rD1F04V3ORD' 291584
    )
    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/rD1F04V4ORD' 291584
    )
    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/rD1F05V1ORD' 291584
    )
    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/rD1F05V2ORD' 291584
    )
    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/rD1F05V3ORD' 291584
    )
    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/rD1F05V4ORD' 291584
)
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/rD1F06V1ORD' 291584
)
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/rD1F06V2ORD' 291584
)
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/rD1F06V3ORD' 291584
)
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/rD1F06V4ORD' 291584
)

```

```

    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/rD1F07V1ORD' 291584
)
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/rD1F07V2ORD' 291584
)
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/rD1F07V3ORD' 291584
)
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/rD1F07V4ORD' 291584
)
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/rD1F08V1ORD' 291584
)
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/rD1F08V2ORD' 291584
)
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/rD1F08V3ORD' 291584
)
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/rD1F08V4ORD' 291584
)
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/rD1F09V1ORD' 291584
)
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/rD1F09V2ORD' 291584
)
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/rD1F09V3ORD' 291584
)
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/rD1F09V4ORD' 291584
)
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/rD1F10V1ORD' 291584
)
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/rD1F10V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_39 of D1

drop tablespace ts_order_39;
create regular tablespace ts_order_39 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_40 of D1

```

```

drop tablespace ts_order_40;
create regular tablespace ts_order_40 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V4ORD' 291584
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

connect reset;

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' 258368
  )
  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' 258368
  )
  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' 258368
  )
  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/rD1F01V4ORDI' 258368
  )
  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/rD1F02V1ORDI' 258368
  )
  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/rD1F02V2ORDI' 258368
  )
  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/rD1F02V3ORDI' 258368
  )
  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/rD1F02V4ORDI' 258368
  )
  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/rD1F03V1ORDI' 258368
  )
  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/rD1F03V2ORDI' 258368
)
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/rD1F03V3ORDI' 258368
)
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/rD1F03V4ORDI' 258368
)
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/rD1F04V1ORDI' 258368
)
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/rD1F04V2ORDI' 258368
)
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/rD1F04V3ORDI' 258368
)
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/rD1F04V4ORDI' 258368
)
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/rD1F05V1ORDI' 258368
)
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/rD1F05V2ORDI' 258368
)
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/rD1F05V3ORDI' 258368
)
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/rD1F05V4ORDI' 258368
)
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/rD1F06V1ORDI' 258368
)
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/rD1F06V2ORDI' 258368
)
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/rD1F06V3ORDI' 258368
)
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/rD1F06V4ORDI' 258368
)
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/rD1F07V1ORDI' 258368

```

```

)
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/rD1F07V2ORDI' 258368
)
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/rD1F07V3ORDI' 258368
)
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/rD1F07V4ORDI' 258368
)
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/rD1F08V1ORDI' 258368
)
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/rD1F08V2ORDI' 258368
)
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/rD1F08V3ORDI' 258368
)
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/rD1F08V4ORDI' 258368
)
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/rD1F09V1ORDI' 258368
)
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/rD1F09V2ORDI' 258368
)
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/rD1F09V3ORDI' 258368
)
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/rD1F09V4ORDI' 258368
)
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/rD1F10V1ORDI' 258368
)
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/rD1F10V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_39 of D1

drop tablespace is_order_39;
create regular tablespace is_order_39 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_40 of D1

drop tablespace is_order_40;
create regular tablespace is_order_40 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V4ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```
connect reset;
```

ts/crts_orderline.ddl

```
connect to tpcc;
```

```
-- 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/rD1F01V1ORL' 7783936
  )
  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/rD1F01V2ORL' 7783936
  )
  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/rD1F01V3ORL' 7783936
  )
  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/rD1F01V4ORL' 7783936
  )
  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/rD1F02V1ORL' 7783936
  )
  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/rD1F02V2ORL' 7783936
  )
  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/rD1F02V3ORL' 7783936
  )
  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/rD1F02V4ORL' 7783936
  )
  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/rD1F03V1ORL' 7783936
  )
  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/rD1F03V2ORL' 7783936
  )
  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/rD1F03V3ORL' 7783936
)
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/rD1F03V4ORL' 7783936
)
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/rD1F04V1ORL' 7783936
)
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/rD1F04V2ORL' 7783936
)
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/rD1F04V3ORL' 7783936
)
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/rD1F04V4ORL' 7783936
)
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/rD1F05V1ORL' 7783936
)
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/rD1F05V2ORL' 7783936
)
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/rD1F05V3ORL' 7783936
)
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/rD1F05V4ORL' 7783936
)
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/rD1F06V1ORL' 7783936
)
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/rD1F06V2ORL' 7783936
)
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/rD1F06V3ORL' 7783936
)
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/rD1F06V4ORL' 7783936
)
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/rD1F07V1ORL' 7783936
)
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/rD1F07V2ORL' 7783936
)
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/rD1F07V3ORL' 7783936
)
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/rD1F07V4ORL' 7783936
)
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/rD1F08V1ORL' 7783936
)
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/rD1F08V2ORL' 7783936
)
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/rD1F08V3ORL' 7783936
)
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/rD1F08V4ORL' 7783936
)
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/rD1F09V1ORL' 7783936
)
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/rD1F09V2ORL' 7783936
)
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/rD1F09V3ORL' 7783936
)
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/rD1F09V4ORL' 7783936

```

```

)
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/rD1F10V1ORL' 7783936
)
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/rD1F10V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_39 of D1

drop tablespace ts_orderline_39;
create regular tablespace ts_orderline_39 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_40 of D1

drop tablespace ts_orderline_40;
create regular tablespace ts_orderline_40 pagesize 8K
managed by database
using
(
    device '/dev/rD1F10V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

ts/crts_stock.ddl

connect to tpcc;

-- now creating TS for ts_stock_01 of D1

```

```

drop tablespace ts_stock_01;
create regular tablespace ts_stock_01 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_02 of D1

drop tablespace ts_stock_02;
create regular tablespace ts_stock_02 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_03 of D1

drop tablespace ts_stock_03;
create regular tablespace ts_stock_03 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_04 of D1

drop tablespace ts_stock_04;
create regular tablespace ts_stock_04 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V4STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_05 of D1

drop tablespace ts_stock_05;
create regular tablespace ts_stock_05 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V1STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_06 of D1

drop tablespace ts_stock_06;
create regular tablespace ts_stock_06 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V2STK' 14001664

```

```

    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_07 of D1

drop tablespace ts_stock_07;
create regular tablespace ts_stock_07 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V3STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_08 of D1

drop tablespace ts_stock_08;
create regular tablespace ts_stock_08 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V4STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_09 of D1

drop tablespace ts_stock_09;
create regular tablespace ts_stock_09 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_10 of D1

drop tablespace ts_stock_10;
create regular tablespace ts_stock_10 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_11 of D1

drop tablespace ts_stock_11;
create regular tablespace ts_stock_11 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_12 of D1

```

```

drop tablespace ts_stock_12;
create regular tablespace ts_stock_12 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_13 of D1

drop tablespace ts_stock_13;
create regular tablespace ts_stock_13 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_14 of D1

drop tablespace ts_stock_14;
create regular tablespace ts_stock_14 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_15 of D1

drop tablespace ts_stock_15;
create regular tablespace ts_stock_15 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_16 of D1

drop tablespace ts_stock_16;
create regular tablespace ts_stock_16 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_17 of D1

drop tablespace ts_stock_17;
create regular tablespace ts_stock_17 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1STK' 14001664

```

```

)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_18 of D1

drop tablespace ts_stock_18;
create regular tablespace ts_stock_18 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_19 of D1

drop tablespace ts_stock_19;
create regular tablespace ts_stock_19 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_20 of D1

drop tablespace ts_stock_20;
create regular tablespace ts_stock_20 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_21 of D1

drop tablespace ts_stock_21;
create regular tablespace ts_stock_21 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_22 of D1

drop tablespace ts_stock_22;
create regular tablespace ts_stock_22 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;

```



```

-- now creating TS for ts_stock_23 of D1

drop tablespace ts_stock_23;
create regular tablespace ts_stock_23 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_24 of D1

drop tablespace ts_stock_24;
create regular tablespace ts_stock_24 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_25 of D1

drop tablespace ts_stock_25;
create regular tablespace ts_stock_25 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_26 of D1

drop tablespace ts_stock_26;
create regular tablespace ts_stock_26 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_27 of D1

drop tablespace ts_stock_27;
create regular tablespace ts_stock_27 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_28 of D1

drop tablespace ts_stock_28;
create regular tablespace ts_stock_28 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F07V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_29 of D1

drop tablespace ts_stock_29;
create regular tablespace ts_stock_29 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_30 of D1

drop tablespace ts_stock_30;
create regular tablespace ts_stock_30 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_31 of D1

drop tablespace ts_stock_31;
create regular tablespace ts_stock_31 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_32 of D1

drop tablespace ts_stock_32;
create regular tablespace ts_stock_32 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_33 of D1

drop tablespace ts_stock_33;
create regular tablespace ts_stock_33 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_34 of D1

```

```

drop tablespace ts_stock_34;
create regular tablespace ts_stock_34 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V2STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

-- now creating TS for ts_stock_35 of D1

```

drop tablespace ts_stock_35;
create regular tablespace ts_stock_35 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V3STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

-- now creating TS for ts_stock_36 of D1

```

drop tablespace ts_stock_36;
create regular tablespace ts_stock_36 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V4STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

-- now creating TS for ts_stock_37 of D1

```

drop tablespace ts_stock_37;
create regular tablespace ts_stock_37 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V1STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

-- now creating TS for ts_stock_38 of D1

```

drop tablespace ts_stock_38;
create regular tablespace ts_stock_38 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V2STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

-- now creating TS for ts_stock_39 of D1

```

drop tablespace ts_stock_39;
create regular tablespace ts_stock_39 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V3STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

    )
    extentsize 256
    prefetchsize 4096;
commit;

```

-- now creating TS for ts_stock_40 of D1

```

drop tablespace ts_stock_40;
create regular tablespace ts_stock_40 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V4STK' 14001664
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

connect reset;

ts/crts_ware.ddl

connect to tpcc;

-- now creating TS for ts_ware_01 of D1

```

drop tablespace ts_ware_01;
create regular tablespace ts_ware_01 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1WARE' 128,
    device '/dev/rD1F01V2WARE' 128,
    device '/dev/rD1F01V3WARE' 128,
    device '/dev/rD1F01V4WARE' 128,
    device '/dev/rD1F02V1WARE' 128
  )
  extentsize 32
  prefetchsize 4096;
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/rD1F02V2WARE' 128,
    device '/dev/rD1F02V3WARE' 128,
    device '/dev/rD1F02V4WARE' 128,
    device '/dev/rD1F03V1WARE' 128,
    device '/dev/rD1F03V2WARE' 128
  )
  extentsize 32
  prefetchsize 4096;
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/rD1F03V3WARE' 128,
    device '/dev/rD1F03V4WARE' 128,
    device '/dev/rD1F04V1WARE' 128,
    device '/dev/rD1F04V2WARE' 128,
    device '/dev/rD1F04V3WARE' 128
  )
  extentsize 32
  prefetchsize 4096;
commit;

```

```

    extentsize 32
    prefetchsize 4096;
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/rD1F04V4WARE' 128,
    device '/dev/rD1F05V1WARE' 128,
    device '/dev/rD1F05V2WARE' 128,
    device '/dev/rD1F05V3WARE' 128,
    device '/dev/rD1F05V4WARE' 128
)
extentsize 32
prefetchsize 4096;
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/rD1F06V1WARE' 128,
    device '/dev/rD1F06V2WARE' 128,
    device '/dev/rD1F06V3WARE' 128,
    device '/dev/rD1F06V4WARE' 128,
    device '/dev/rD1F07V1WARE' 128
)
extentsize 32
prefetchsize 4096;
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/rD1F07V2WARE' 128,
    device '/dev/rD1F07V3WARE' 128,
    device '/dev/rD1F07V4WARE' 128,
    device '/dev/rD1F08V1WARE' 128,
    device '/dev/rD1F08V2WARE' 128
)
extentsize 32
prefetchsize 4096;
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/rD1F08V3WARE' 128,
    device '/dev/rD1F08V4WARE' 128,
    device '/dev/rD1F09V1WARE' 128,
    device '/dev/rD1F09V2WARE' 128,
    device '/dev/rD1F09V3WARE' 128
)
extentsize 32
prefetchsize 4096;
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/rD1F09V4WARE' 128,
    device '/dev/rD1F10V1WARE' 128,
    device '/dev/rD1F10V2WARE' 128,
    device '/dev/rD1F10V3WARE' 128,
    device '/dev/rD1F10V4WARE' 128
)
extentsize 32
prefetchsize 4096;
commit;

connect reset;

```

bp/alter bufferpool.ddl

```

connect to tpcc;

alter bufferpool IBMDEFAULTBP DEFERRED size 500;
alter bufferpool IBMDEFAULTBP8K DEFERRED size 100;
alter bufferpool IBMDEFAULTBP16K DEFERRED size 100;

```

```

alter bufferpool WAR DEFERRED size 1750;
alter bufferpool DIS DEFERRED size 18500;
alter bufferpool ITM DEFERRED size 1500;
alter bufferpool HST DEFERRED size 3000;
alter bufferpool NEW DEFERRED size 1950000;
alter bufferpool ORD DEFERRED size 1400000;
alter bufferpool ORD_I DEFERRED size 4000000;
alter bufferpool OLN DEFERRED size 2600000;
alter bufferpool CST DEFERRED size 5750000;
alter bufferpool CST_I DEFERRED size 2000000;
alter bufferpool STK DEFERRED size 78000000;

```

```
connect reset;
```

bp/create bufferpool.ddl

```
connect to tpcc;
```

```

create bufferpool IBMDEFAULTBP8K size 1000 pagesize 8192;
create bufferpool IBMDEFAULTBP16K size 1000 pagesize 16384;

```

```

create bufferpool WAR size 1000 pagesize 4096;
create bufferpool DIS size 1000 pagesize 4096;
create bufferpool ITM size 1000 pagesize 8192;
create bufferpool HST size 1000 pagesize 16384;
create bufferpool NEW size 1000 pagesize 4096;
create bufferpool ORD size 1000 pagesize 8192;
create bufferpool ORD_I size 1000 pagesize 8192;
create bufferpool OLN size 1000 pagesize 8192;
create bufferpool CST size 1000 pagesize 4096;
create bufferpool CST_I size 1000 pagesize 8192;
create bufferpool STK size 100000 pagesize 4096;

```

```
connect reset;
```

bp/alter tablespace.ddl


```

alter tablespace ts_stock_37 prefetchsize 4096;
alter tablespace ts_stock_38 prefetchsize 4096;
alter tablespace ts_stock_39 prefetchsize 4096;
alter tablespace ts_stock_40 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;
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 1600);
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 1601 AND 3200);
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 3201 AND 4800);
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 4801 AND 6400);
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 6401 AND 8000);
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 8001 AND 9600);

```

```

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 9601 AND 11200);
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 11201 AND
12800);
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 12801 AND
14400);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER10 OFF;
ALTER TABLE CUSTOMER10 DROP CONSTRAINT
CUSTOMER10CKC;
ALTER TABLE CUSTOMER10 ADD CONSTRAINT
CUSTOMER10CKC CHECK (C_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER11 OFF;
ALTER TABLE CUSTOMER11 DROP CONSTRAINT
CUSTOMER11CKC;
ALTER TABLE CUSTOMER11 ADD CONSTRAINT
CUSTOMER11CKC CHECK (C_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER12 OFF;
ALTER TABLE CUSTOMER12 DROP CONSTRAINT
CUSTOMER12CKC;
ALTER TABLE CUSTOMER12 ADD CONSTRAINT
CUSTOMER12CKC CHECK (C_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER13 OFF;
ALTER TABLE CUSTOMER13 DROP CONSTRAINT
CUSTOMER13CKC;
ALTER TABLE CUSTOMER13 ADD CONSTRAINT
CUSTOMER13CKC CHECK (C_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER14 OFF;
ALTER TABLE CUSTOMER14 DROP CONSTRAINT
CUSTOMER14CKC;
ALTER TABLE CUSTOMER14 ADD CONSTRAINT
CUSTOMER14CKC CHECK (C_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER15 OFF;
ALTER TABLE CUSTOMER15 DROP CONSTRAINT
CUSTOMER15CKC;
ALTER TABLE CUSTOMER15 ADD CONSTRAINT
CUSTOMER15CKC CHECK (C_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER16 OFF;
ALTER TABLE CUSTOMER16 DROP CONSTRAINT
CUSTOMER16CKC;
ALTER TABLE CUSTOMER16 ADD CONSTRAINT
CUSTOMER16CKC CHECK (C_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER17 OFF;
ALTER TABLE CUSTOMER17 DROP CONSTRAINT
CUSTOMER17CKC;
ALTER TABLE CUSTOMER17 ADD CONSTRAINT
CUSTOMER17CKC CHECK (C_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER18 OFF;
ALTER TABLE CUSTOMER18 DROP CONSTRAINT
CUSTOMER18CKC;
ALTER TABLE CUSTOMER18 ADD CONSTRAINT
CUSTOMER18CKC CHECK (C_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER19 OFF;
ALTER TABLE CUSTOMER19 DROP CONSTRAINT
CUSTOMER19CKC;
ALTER TABLE CUSTOMER19 ADD CONSTRAINT
CUSTOMER19CKC CHECK (C_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER20 OFF;
ALTER TABLE CUSTOMER20 DROP CONSTRAINT
CUSTOMER20CKC;
ALTER TABLE CUSTOMER20 ADD CONSTRAINT
CUSTOMER20CKC CHECK (C_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER21 OFF;

```

```

ALTER TABLE CUSTOMER21 DROP CONSTRAINT
CUSTOMER21CKC;
ALTER TABLE CUSTOMER21 ADD CONSTRAINT
CUSTOMER21CKC CHECK (C_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER22 OFF;
ALTER TABLE CUSTOMER22 DROP CONSTRAINT
CUSTOMER22CKC;
ALTER TABLE CUSTOMER22 ADD CONSTRAINT
CUSTOMER22CKC CHECK (C_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER23 OFF;
ALTER TABLE CUSTOMER23 DROP CONSTRAINT
CUSTOMER23CKC;
ALTER TABLE CUSTOMER23 ADD CONSTRAINT
CUSTOMER23CKC CHECK (C_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER24 OFF;
ALTER TABLE CUSTOMER24 DROP CONSTRAINT
CUSTOMER24CKC;
ALTER TABLE CUSTOMER24 ADD CONSTRAINT
CUSTOMER24CKC CHECK (C_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER25 OFF;
ALTER TABLE CUSTOMER25 DROP CONSTRAINT
CUSTOMER25CKC;
ALTER TABLE CUSTOMER25 ADD CONSTRAINT
CUSTOMER25CKC CHECK (C_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER26 OFF;
ALTER TABLE CUSTOMER26 DROP CONSTRAINT
CUSTOMER26CKC;
ALTER TABLE CUSTOMER26 ADD CONSTRAINT
CUSTOMER26CKC CHECK (C_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER27 OFF;
ALTER TABLE CUSTOMER27 DROP CONSTRAINT
CUSTOMER27CKC;
ALTER TABLE CUSTOMER27 ADD CONSTRAINT
CUSTOMER27CKC CHECK (C_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER28 OFF;
ALTER TABLE CUSTOMER28 DROP CONSTRAINT
CUSTOMER28CKC;

```

```

ALTER TABLE CUSTOMER28 ADD CONSTRAINT
CUSTOMER28CKC CHECK (C_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER29 OFF;
ALTER TABLE CUSTOMER29 DROP CONSTRAINT
CUSTOMER29CKC;
ALTER TABLE CUSTOMER29 ADD CONSTRAINT
CUSTOMER29CKC CHECK (C_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER30 OFF;
ALTER TABLE CUSTOMER30 DROP CONSTRAINT
CUSTOMER30CKC;
ALTER TABLE CUSTOMER30 ADD CONSTRAINT
CUSTOMER30CKC CHECK (C_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER31 OFF;
ALTER TABLE CUSTOMER31 DROP CONSTRAINT
CUSTOMER31CKC;
ALTER TABLE CUSTOMER31 ADD CONSTRAINT
CUSTOMER31CKC CHECK (C_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER32 OFF;
ALTER TABLE CUSTOMER32 DROP CONSTRAINT
CUSTOMER32CKC;
ALTER TABLE CUSTOMER32 ADD CONSTRAINT
CUSTOMER32CKC CHECK (C_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT
CUSTOMER33CKC;
ALTER TABLE CUSTOMER33 ADD CONSTRAINT
CUSTOMER33CKC CHECK (C_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT
CUSTOMER34CKC;
ALTER TABLE CUSTOMER34 ADD CONSTRAINT
CUSTOMER34CKC CHECK (C_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT
CUSTOMER35CKC;

```

```

ALTER TABLE CUSTOMER35 ADD CONSTRAINT
CUSTOMER35CKC CHECK (C_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR CUSTOMER35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER36 OFF;
ALTER TABLE CUSTOMER36 DROP CONSTRAINT
CUSTOMER36CKC;
ALTER TABLE CUSTOMER36 ADD CONSTRAINT
CUSTOMER36CKC CHECK (C_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR CUSTOMER36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER37 OFF;
ALTER TABLE CUSTOMER37 DROP CONSTRAINT
CUSTOMER37CKC;
ALTER TABLE CUSTOMER37 ADD CONSTRAINT
CUSTOMER37CKC CHECK (C_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR CUSTOMER37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER38 OFF;
ALTER TABLE CUSTOMER38 DROP CONSTRAINT
CUSTOMER38CKC;
ALTER TABLE CUSTOMER38 ADD CONSTRAINT
CUSTOMER38CKC CHECK (C_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR CUSTOMER38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER39 OFF;
ALTER TABLE CUSTOMER39 DROP CONSTRAINT
CUSTOMER39CKC;
ALTER TABLE CUSTOMER39 ADD CONSTRAINT
CUSTOMER39CKC CHECK (C_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR CUSTOMER39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER40 OFF;
ALTER TABLE CUSTOMER40 DROP CONSTRAINT
CUSTOMER40CKC;
ALTER TABLE CUSTOMER40 ADD CONSTRAINT
CUSTOMER40CKC CHECK (C_W_ID >= 62401);
SET INTEGRITY FOR CUSTOMER40 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 8000);
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 8001 AND 16000);
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 16001 AND 24000);
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 24001 AND 32000);
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 32001 AND 40000);
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 40001 AND 48000);
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 48001 AND 56000);
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 >= 56001);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

CRCONST HISTORY.ddl

```

connect to TPCC in share mode;

SET INTEGRITY FOR HISTORY1 OFF;
ALTER TABLE HISTORY1 DROP CONSTRAINT HISTORY1CKC;
ALTER TABLE HISTORY1 ADD CONSTRAINT HISTORY1CKC
CHECK (H_W_ID BETWEEN 1 AND 8000);
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 8001 AND 16000);
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 16001 AND 24000);
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 24001 AND 32000);
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 32001 AND 40000);
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 40001 AND 48000);
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 48001 AND 56000);
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 >= 56001);
SET INTEGRITY FOR HISTORY8 ALL IMMEDIATE UNCHECKED;
connect reset;

```

CRCONST NEW ORDER.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA1 OFF;
ALTER TABLE NEW_ORDERA1 DROP CONSTRAINT
NEW_ORDERA1CKC;
ALTER TABLE NEW_ORDERA1 ADD CONSTRAINT
NEW_ORDERA1CKC CHECK ((NO_W_ID BETWEEN 1 AND
8000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA2 OFF;
ALTER TABLE NEW_ORDERA2 DROP CONSTRAINT
NEW_ORDERA2CKC;
ALTER TABLE NEW_ORDERA2 ADD CONSTRAINT
NEW_ORDERA2CKC CHECK ((NO_W_ID BETWEEN 8001 AND
16000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA3 OFF;
ALTER TABLE NEW_ORDERA3 DROP CONSTRAINT
NEW_ORDERA3CKC;
ALTER TABLE NEW_ORDERA3 ADD CONSTRAINT
NEW_ORDERA3CKC CHECK ((NO_W_ID BETWEEN 16001 AND
24000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE
UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA4 OFF;
ALTER TABLE NEW_ORDERA4 DROP CONSTRAINT
NEW_ORDERA4CKC;
ALTER TABLE NEW_ORDERA4 ADD CONSTRAINT
NEW_ORDERA4CKC CHECK ((NO_W_ID BETWEEN 24001 AND
32000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA5 OFF;
ALTER TABLE NEW_ORDERA5 DROP CONSTRAINT
NEW_ORDERA5CKC;
ALTER TABLE NEW_ORDERA5 ADD CONSTRAINT
NEW_ORDERA5CKC CHECK ((NO_W_ID BETWEEN 32001 AND
40000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA6 OFF;
ALTER TABLE NEW_ORDERA6 DROP CONSTRAINT
NEW_ORDERA6CKC;
ALTER TABLE NEW_ORDERA6 ADD CONSTRAINT
NEW_ORDERA6CKC CHECK ((NO_W_ID BETWEEN 40001 AND
48000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA7 OFF;
ALTER TABLE NEW_ORDERA7 DROP CONSTRAINT
NEW_ORDERA7CKC;
ALTER TABLE NEW_ORDERA7 ADD CONSTRAINT
NEW_ORDERA7CKC CHECK ((NO_W_ID BETWEEN 48001 AND
56000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA8 OFF;
ALTER TABLE NEW_ORDERA8 DROP CONSTRAINT
NEW_ORDERA8CKC;
ALTER TABLE NEW_ORDERA8 ADD CONSTRAINT
NEW_ORDERA8CKC CHECK ((NO_W_ID >= 56001) AND
(NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB1 OFF;
ALTER TABLE NEW_ORDERB1 DROP CONSTRAINT
NEW_ORDERB1CKC;
ALTER TABLE NEW_ORDERB1 ADD CONSTRAINT
NEW_ORDERB1CKC CHECK ((NO_W_ID BETWEEN 1 AND
8000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB2 OFF;
ALTER TABLE NEW_ORDERB2 DROP CONSTRAINT
NEW_ORDERB2CKC;
ALTER TABLE NEW_ORDERB2 ADD CONSTRAINT
NEW_ORDERB2CKC CHECK ((NO_W_ID BETWEEN 8001 AND
16000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;

```

```

SET INTEGRITY FOR NEW_ORDERB3 OFF;
ALTER TABLE NEW_ORDERB3 DROP CONSTRAINT
NEW_ORDERB3CKC;
ALTER TABLE NEW_ORDERB3 ADD CONSTRAINT
NEW_ORDERB3CKC CHECK ((NO_W_ID BETWEEN 16001 AND
24000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB4 OFF;
ALTER TABLE NEW_ORDERB4 DROP CONSTRAINT
NEW_ORDERB4CKC;
ALTER TABLE NEW_ORDERB4 ADD CONSTRAINT
NEW_ORDERB4CKC CHECK ((NO_W_ID BETWEEN 24001 AND
32000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB5 OFF;
ALTER TABLE NEW_ORDERB5 DROP CONSTRAINT
NEW_ORDERB5CKC;
ALTER TABLE NEW_ORDERB5 ADD CONSTRAINT
NEW_ORDERB5CKC CHECK ((NO_W_ID BETWEEN 32001 AND
40000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB6 OFF;
ALTER TABLE NEW_ORDERB6 DROP CONSTRAINT
NEW_ORDERB6CKC;
ALTER TABLE NEW_ORDERB6 ADD CONSTRAINT
NEW_ORDERB6CKC CHECK ((NO_W_ID BETWEEN 40001 AND
48000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB7 OFF;
ALTER TABLE NEW_ORDERB7 DROP CONSTRAINT
NEW_ORDERB7CKC;
ALTER TABLE NEW_ORDERB7 ADD CONSTRAINT
NEW_ORDERB7CKC CHECK ((NO_W_ID BETWEEN 48001 AND
56000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB8 OFF;
ALTER TABLE NEW_ORDERB8 DROP CONSTRAINT
NEW_ORDERB8CKC;
ALTER TABLE NEW_ORDERB8 ADD CONSTRAINT
NEW_ORDERB8CKC CHECK ((NO_W_ID >= 56001) AND
(NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB8 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST ORDERS.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS1 OFF;
ALTER TABLE ORDERS1 DROP CONSTRAINT ORDERS1CKC;
ALTER TABLE ORDERS1 ADD CONSTRAINT ORDERS1CKC
CHECK (O_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR ORDERS1 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS2 OFF;
ALTER TABLE ORDERS2 DROP CONSTRAINT ORDERS2CKC;
ALTER TABLE ORDERS2 ADD CONSTRAINT ORDERS2CKC
CHECK (O_W_ID BETWEEN 1601 AND 3200);
SET INTEGRITY FOR ORDERS2 ALL IMMEDIATE UNCHECKED;

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS3 OFF;
ALTER TABLE ORDERS3 DROP CONSTRAINT ORDERS3CKC;
ALTER TABLE ORDERS3 ADD CONSTRAINT ORDERS3CKC
CHECK (O_W_ID BETWEEN 3201 AND 4800);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS4 OFF;
ALTER TABLE ORDERS4 DROP CONSTRAINT ORDERS4CKC;
ALTER TABLE ORDERS4 ADD CONSTRAINT ORDERS4CKC
CHECK (O_W_ID BETWEEN 4801 AND 6400);
SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS5 OFF;
ALTER TABLE ORDERS5 DROP CONSTRAINT ORDERS5CKC;
ALTER TABLE ORDERS5 ADD CONSTRAINT ORDERS5CKC
CHECK (O_W_ID BETWEEN 6401 AND 8000);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS6 OFF;
ALTER TABLE ORDERS6 DROP CONSTRAINT ORDERS6CKC;
ALTER TABLE ORDERS6 ADD CONSTRAINT ORDERS6CKC
CHECK (O_W_ID BETWEEN 8001 AND 9600);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS7 OFF;
ALTER TABLE ORDERS7 DROP CONSTRAINT ORDERS7CKC;
ALTER TABLE ORDERS7 ADD CONSTRAINT ORDERS7CKC
CHECK (O_W_ID BETWEEN 9601 AND 11200);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS8 OFF;
ALTER TABLE ORDERS8 DROP CONSTRAINT ORDERS8CKC;
ALTER TABLE ORDERS8 ADD CONSTRAINT ORDERS8CKC
CHECK (O_W_ID BETWEEN 11201 AND 12800);
SET INTEGRITY FOR ORDERS8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS9 OFF;
ALTER TABLE ORDERS9 DROP CONSTRAINT ORDERS9CKC;
ALTER TABLE ORDERS9 ADD CONSTRAINT ORDERS9CKC
CHECK (O_W_ID BETWEEN 12801 AND 14400);
SET INTEGRITY FOR ORDERS9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS10 OFF;
ALTER TABLE ORDERS10 DROP CONSTRAINT ORDERS10CKC;
ALTER TABLE ORDERS10 ADD CONSTRAINT ORDERS10CKC
CHECK (O_W_ID BETWEEN 14401 AND 16000);
SET INTEGRITY FOR ORDERS10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS11 OFF;
ALTER TABLE ORDERS11 DROP CONSTRAINT ORDERS11CKC;
ALTER TABLE ORDERS11 ADD CONSTRAINT ORDERS11CKC
CHECK (O_W_ID BETWEEN 16001 AND 17600);
SET INTEGRITY FOR ORDERS11 ALL IMMEDIATE
UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS12 OFF;
ALTER TABLE ORDERS12 DROP CONSTRAINT ORDERS12CKC;
ALTER TABLE ORDERS12 ADD CONSTRAINT ORDERS12CKC
CHECK (O_W_ID BETWEEN 17601 AND 19200);
SET INTEGRITY FOR ORDERS12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS13 OFF;
ALTER TABLE ORDERS13 DROP CONSTRAINT ORDERS13CKC;
ALTER TABLE ORDERS13 ADD CONSTRAINT ORDERS13CKC
CHECK (O_W_ID BETWEEN 19201 AND 20800);
SET INTEGRITY FOR ORDERS13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS14 OFF;
ALTER TABLE ORDERS14 DROP CONSTRAINT ORDERS14CKC;
ALTER TABLE ORDERS14 ADD CONSTRAINT ORDERS14CKC
CHECK (O_W_ID BETWEEN 20801 AND 22400);
SET INTEGRITY FOR ORDERS14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS15 OFF;
ALTER TABLE ORDERS15 DROP CONSTRAINT ORDERS15CKC;
ALTER TABLE ORDERS15 ADD CONSTRAINT ORDERS15CKC
CHECK (O_W_ID BETWEEN 22401 AND 24000);
SET INTEGRITY FOR ORDERS15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS16 OFF;
ALTER TABLE ORDERS16 DROP CONSTRAINT ORDERS16CKC;
ALTER TABLE ORDERS16 ADD CONSTRAINT ORDERS16CKC
CHECK (O_W_ID BETWEEN 24001 AND 25600);
SET INTEGRITY FOR ORDERS16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS17 OFF;
ALTER TABLE ORDERS17 DROP CONSTRAINT ORDERS17CKC;
ALTER TABLE ORDERS17 ADD CONSTRAINT ORDERS17CKC
CHECK (O_W_ID BETWEEN 25601 AND 27200);
SET INTEGRITY FOR ORDERS17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS18 OFF;
ALTER TABLE ORDERS18 DROP CONSTRAINT ORDERS18CKC;
ALTER TABLE ORDERS18 ADD CONSTRAINT ORDERS18CKC
CHECK (O_W_ID BETWEEN 27201 AND 28800);
SET INTEGRITY FOR ORDERS18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS19 OFF;
ALTER TABLE ORDERS19 DROP CONSTRAINT ORDERS19CKC;
ALTER TABLE ORDERS19 ADD CONSTRAINT ORDERS19CKC
CHECK (O_W_ID BETWEEN 28801 AND 30400);
SET INTEGRITY FOR ORDERS19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS20 OFF;
ALTER TABLE ORDERS20 DROP CONSTRAINT ORDERS20CKC;
ALTER TABLE ORDERS20 ADD CONSTRAINT ORDERS20CKC
CHECK (O_W_ID BETWEEN 30401 AND 32000);
SET INTEGRITY FOR ORDERS20 ALL IMMEDIATE
UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS21 OFF;
ALTER TABLE ORDERS21 DROP CONSTRAINT ORDERS21CKC;
ALTER TABLE ORDERS21 ADD CONSTRAINT ORDERS21CKC
CHECK (O_W_ID BETWEEN 32001 AND 33600);
SET INTEGRITY FOR ORDERS21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS22 OFF;
ALTER TABLE ORDERS22 DROP CONSTRAINT ORDERS22CKC;
ALTER TABLE ORDERS22 ADD CONSTRAINT ORDERS22CKC
CHECK (O_W_ID BETWEEN 33601 AND 35200);

SET INTEGRITY FOR ORDERS22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS23 OFF;
ALTER TABLE ORDERS23 DROP CONSTRAINT ORDERS23CKC;
ALTER TABLE ORDERS23 ADD CONSTRAINT ORDERS23CKC
CHECK (O_W_ID BETWEEN 35201 AND 36800);
SET INTEGRITY FOR ORDERS23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS24 OFF;
ALTER TABLE ORDERS24 DROP CONSTRAINT ORDERS24CKC;
ALTER TABLE ORDERS24 ADD CONSTRAINT ORDERS24CKC
CHECK (O_W_ID BETWEEN 36801 AND 38400);
SET INTEGRITY FOR ORDERS24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS25 OFF;
ALTER TABLE ORDERS25 DROP CONSTRAINT ORDERS25CKC;
ALTER TABLE ORDERS25 ADD CONSTRAINT ORDERS25CKC
CHECK (O_W_ID BETWEEN 38401 AND 40000);
SET INTEGRITY FOR ORDERS25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS26 OFF;
ALTER TABLE ORDERS26 DROP CONSTRAINT ORDERS26CKC;
ALTER TABLE ORDERS26 ADD CONSTRAINT ORDERS26CKC
CHECK (O_W_ID BETWEEN 40001 AND 41600);
SET INTEGRITY FOR ORDERS26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS27 OFF;
ALTER TABLE ORDERS27 DROP CONSTRAINT ORDERS27CKC;
ALTER TABLE ORDERS27 ADD CONSTRAINT ORDERS27CKC
CHECK (O_W_ID BETWEEN 41601 AND 43200);
SET INTEGRITY FOR ORDERS27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS28 OFF;
ALTER TABLE ORDERS28 DROP CONSTRAINT ORDERS28CKC;
ALTER TABLE ORDERS28 ADD CONSTRAINT ORDERS28CKC
CHECK (O_W_ID BETWEEN 43201 AND 44800);
SET INTEGRITY FOR ORDERS28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS29 OFF;
ALTER TABLE ORDERS29 DROP CONSTRAINT ORDERS29CKC;
ALTER TABLE ORDERS29 ADD CONSTRAINT ORDERS29CKC
CHECK (O_W_ID BETWEEN 44801 AND 46400);

```

```

SET INTEGRITY FOR ORDERS29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS30 OFF;
ALTER TABLE ORDERS30 DROP CONSTRAINT ORDERS30CKC;
ALTER TABLE ORDERS30 ADD CONSTRAINT ORDERS30CKC
CHECK (O_W_ID BETWEEN 46401 AND 48000);
SET INTEGRITY FOR ORDERS30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS31 OFF;
ALTER TABLE ORDERS31 DROP CONSTRAINT ORDERS31CKC;
ALTER TABLE ORDERS31 ADD CONSTRAINT ORDERS31CKC
CHECK (O_W_ID BETWEEN 48001 AND 49600);
SET INTEGRITY FOR ORDERS31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS32 OFF;
ALTER TABLE ORDERS32 DROP CONSTRAINT ORDERS32CKC;
ALTER TABLE ORDERS32 ADD CONSTRAINT ORDERS32CKC
CHECK (O_W_ID BETWEEN 49601 AND 51200);
SET INTEGRITY FOR ORDERS32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS33 OFF;
ALTER TABLE ORDERS33 DROP CONSTRAINT ORDERS33CKC;
ALTER TABLE ORDERS33 ADD CONSTRAINT ORDERS33CKC
CHECK (O_W_ID BETWEEN 51201 AND 52800);
SET INTEGRITY FOR ORDERS33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS34 OFF;
ALTER TABLE ORDERS34 DROP CONSTRAINT ORDERS34CKC;
ALTER TABLE ORDERS34 ADD CONSTRAINT ORDERS34CKC
CHECK (O_W_ID BETWEEN 52801 AND 54400);
SET INTEGRITY FOR ORDERS34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS35 OFF;
ALTER TABLE ORDERS35 DROP CONSTRAINT ORDERS35CKC;
ALTER TABLE ORDERS35 ADD CONSTRAINT ORDERS35CKC
CHECK (O_W_ID BETWEEN 54401 AND 56000);
SET INTEGRITY FOR ORDERS35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS36 OFF;
ALTER TABLE ORDERS36 DROP CONSTRAINT ORDERS36CKC;
ALTER TABLE ORDERS36 ADD CONSTRAINT ORDERS36CKC
CHECK (O_W_ID BETWEEN 56001 AND 57600);
SET INTEGRITY FOR ORDERS36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS37 OFF;
ALTER TABLE ORDERS37 DROP CONSTRAINT ORDERS37CKC;
ALTER TABLE ORDERS37 ADD CONSTRAINT ORDERS37CKC
CHECK (O_W_ID BETWEEN 57601 AND 59200);
SET INTEGRITY FOR ORDERS37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS38 OFF;
ALTER TABLE ORDERS38 DROP CONSTRAINT ORDERS38CKC;
ALTER TABLE ORDERS38 ADD CONSTRAINT ORDERS38CKC
CHECK (O_W_ID BETWEEN 59201 AND 60800);

```



```

SET INTEGRITY FOR ORDERS38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS39 OFF;
ALTER TABLE ORDERS39 DROP CONSTRAINT ORDERS39CKC;
ALTER TABLE ORDERS39 ADD CONSTRAINT ORDERS39CKC
CHECK (O_W_ID BETWEEN 60801 AND 62400);
SET INTEGRITY FOR ORDERS39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS40 OFF;
ALTER TABLE ORDERS40 DROP CONSTRAINT ORDERS40CKC;
ALTER TABLE ORDERS40 ADD CONSTRAINT ORDERS40CKC
CHECK (O_W_ID >= 62401);
SET INTEGRITY FOR ORDERS40 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST ORDER_LINE.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE1 OFF;
ALTER TABLE ORDER_LINE1 DROP CONSTRAINT
ORDER_LINE1CKC;
ALTER TABLE ORDER_LINE1 ADD CONSTRAINT
ORDER_LINE1CKC CHECK (OL_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE2 OFF;
ALTER TABLE ORDER_LINE2 DROP CONSTRAINT
ORDER_LINE2CKC;
ALTER TABLE ORDER_LINE2 ADD CONSTRAINT
ORDER_LINE2CKC CHECK (OL_W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE3 OFF;
ALTER TABLE ORDER_LINE3 DROP CONSTRAINT
ORDER_LINE3CKC;
ALTER TABLE ORDER_LINE3 ADD CONSTRAINT
ORDER_LINE3CKC CHECK (OL_W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE4 OFF;
ALTER TABLE ORDER_LINE4 DROP CONSTRAINT
ORDER_LINE4CKC;
ALTER TABLE ORDER_LINE4 ADD CONSTRAINT
ORDER_LINE4CKC CHECK (OL_W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE5 OFF;
ALTER TABLE ORDER_LINE5 DROP CONSTRAINT
ORDER_LINE5CKC;
ALTER TABLE ORDER_LINE5 ADD CONSTRAINT
ORDER_LINE5CKC CHECK (OL_W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE
UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE6 OFF;
ALTER TABLE ORDER_LINE6 DROP CONSTRAINT
ORDER_LINE6CKC;
ALTER TABLE ORDER_LINE6 ADD CONSTRAINT
ORDER_LINE6CKC CHECK (OL_W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE7 OFF;
ALTER TABLE ORDER_LINE7 DROP CONSTRAINT
ORDER_LINE7CKC;
ALTER TABLE ORDER_LINE7 ADD CONSTRAINT
ORDER_LINE7CKC CHECK (OL_W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE8 OFF;
ALTER TABLE ORDER_LINE8 DROP CONSTRAINT
ORDER_LINE8CKC;
ALTER TABLE ORDER_LINE8 ADD CONSTRAINT
ORDER_LINE8CKC CHECK (OL_W_ID BETWEEN 11201 AND
12800);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE9 OFF;
ALTER TABLE ORDER_LINE9 DROP CONSTRAINT
ORDER_LINE9CKC;
ALTER TABLE ORDER_LINE9 ADD CONSTRAINT
ORDER_LINE9CKC CHECK (OL_W_ID BETWEEN 12801 AND
14400);
SET INTEGRITY FOR ORDER_LINE9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE10 OFF;
ALTER TABLE ORDER_LINE10 DROP CONSTRAINT
ORDER_LINE10CKC;
ALTER TABLE ORDER_LINE10 ADD CONSTRAINT
ORDER_LINE10CKC CHECK (OL_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR ORDER_LINE10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE11 OFF;
ALTER TABLE ORDER_LINE11 DROP CONSTRAINT
ORDER_LINE11CKC;
ALTER TABLE ORDER_LINE11 ADD CONSTRAINT
ORDER_LINE11CKC CHECK (OL_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR ORDER_LINE11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE12 OFF;
ALTER TABLE ORDER_LINE12 DROP CONSTRAINT
ORDER_LINE12CKC;
ALTER TABLE ORDER_LINE12 ADD CONSTRAINT
ORDER_LINE12CKC CHECK (OL_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR ORDER_LINE12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;

```

```

SET INTEGRITY FOR ORDER_LINE13 OFF;
ALTER TABLE ORDER_LINE13 DROP CONSTRAINT
ORDER_LINE13CKC;
ALTER TABLE ORDER_LINE13 ADD CONSTRAINT
ORDER_LINE13CKC CHECK (OL_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR ORDER_LINE13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE14 OFF;
ALTER TABLE ORDER_LINE14 DROP CONSTRAINT
ORDER_LINE14CKC;
ALTER TABLE ORDER_LINE14 ADD CONSTRAINT
ORDER_LINE14CKC CHECK (OL_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR ORDER_LINE14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE15 OFF;
ALTER TABLE ORDER_LINE15 DROP CONSTRAINT
ORDER_LINE15CKC;
ALTER TABLE ORDER_LINE15 ADD CONSTRAINT
ORDER_LINE15CKC CHECK (OL_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR ORDER_LINE15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE16 OFF;
ALTER TABLE ORDER_LINE16 DROP CONSTRAINT
ORDER_LINE16CKC;
ALTER TABLE ORDER_LINE16 ADD CONSTRAINT
ORDER_LINE16CKC CHECK (OL_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR ORDER_LINE16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE17 OFF;
ALTER TABLE ORDER_LINE17 DROP CONSTRAINT
ORDER_LINE17CKC;
ALTER TABLE ORDER_LINE17 ADD CONSTRAINT
ORDER_LINE17CKC CHECK (OL_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR ORDER_LINE17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE18 OFF;
ALTER TABLE ORDER_LINE18 DROP CONSTRAINT
ORDER_LINE18CKC;
ALTER TABLE ORDER_LINE18 ADD CONSTRAINT
ORDER_LINE18CKC CHECK (OL_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR ORDER_LINE18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE19 OFF;
ALTER TABLE ORDER_LINE19 DROP CONSTRAINT
ORDER_LINE19CKC;
ALTER TABLE ORDER_LINE19 ADD CONSTRAINT
ORDER_LINE19CKC CHECK (OL_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR ORDER_LINE19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE20 OFF;

```

```

ALTER TABLE ORDER_LINE20 DROP CONSTRAINT
ORDER_LINE20CKC;
ALTER TABLE ORDER_LINE20 ADD CONSTRAINT
ORDER_LINE20CKC CHECK (OL_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR ORDER_LINE20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE21 OFF;
ALTER TABLE ORDER_LINE21 DROP CONSTRAINT
ORDER_LINE21CKC;
ALTER TABLE ORDER_LINE21 ADD CONSTRAINT
ORDER_LINE21CKC CHECK (OL_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR ORDER_LINE21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE22 OFF;
ALTER TABLE ORDER_LINE22 DROP CONSTRAINT
ORDER_LINE22CKC;
ALTER TABLE ORDER_LINE22 ADD CONSTRAINT
ORDER_LINE22CKC CHECK (OL_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR ORDER_LINE22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE23 OFF;
ALTER TABLE ORDER_LINE23 DROP CONSTRAINT
ORDER_LINE23CKC;
ALTER TABLE ORDER_LINE23 ADD CONSTRAINT
ORDER_LINE23CKC CHECK (OL_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR ORDER_LINE23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE24 OFF;
ALTER TABLE ORDER_LINE24 DROP CONSTRAINT
ORDER_LINE24CKC;
ALTER TABLE ORDER_LINE24 ADD CONSTRAINT
ORDER_LINE24CKC CHECK (OL_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR ORDER_LINE24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE25 OFF;
ALTER TABLE ORDER_LINE25 DROP CONSTRAINT
ORDER_LINE25CKC;
ALTER TABLE ORDER_LINE25 ADD CONSTRAINT
ORDER_LINE25CKC CHECK (OL_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR ORDER_LINE25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE26 OFF;
ALTER TABLE ORDER_LINE26 DROP CONSTRAINT
ORDER_LINE26CKC;
ALTER TABLE ORDER_LINE26 ADD CONSTRAINT
ORDER_LINE26CKC CHECK (OL_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR ORDER_LINE26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE27 OFF;
ALTER TABLE ORDER_LINE27 DROP CONSTRAINT
ORDER_LINE27CKC;

```

```

ALTER TABLE ORDER_LINE27 ADD CONSTRAINT
ORDER_LINE27CKC CHECK (OL_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR ORDER_LINE27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE28 OFF;
ALTER TABLE ORDER_LINE28 DROP CONSTRAINT
ORDER_LINE28CKC;
ALTER TABLE ORDER_LINE28 ADD CONSTRAINT
ORDER_LINE28CKC CHECK (OL_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE29 OFF;
ALTER TABLE ORDER_LINE29 DROP CONSTRAINT
ORDER_LINE29CKC;
ALTER TABLE ORDER_LINE29 ADD CONSTRAINT
ORDER_LINE29CKC CHECK (OL_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR ORDER_LINE29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE30 OFF;
ALTER TABLE ORDER_LINE30 DROP CONSTRAINT
ORDER_LINE30CKC;
ALTER TABLE ORDER_LINE30 ADD CONSTRAINT
ORDER_LINE30CKC CHECK (OL_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR ORDER_LINE30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE31 OFF;
ALTER TABLE ORDER_LINE31 DROP CONSTRAINT
ORDER_LINE31CKC;
ALTER TABLE ORDER_LINE31 ADD CONSTRAINT
ORDER_LINE31CKC CHECK (OL_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR ORDER_LINE31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE32 OFF;
ALTER TABLE ORDER_LINE32 DROP CONSTRAINT
ORDER_LINE32CKC;
ALTER TABLE ORDER_LINE32 ADD CONSTRAINT
ORDER_LINE32CKC CHECK (OL_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE33 OFF;
ALTER TABLE ORDER_LINE33 DROP CONSTRAINT
ORDER_LINE33CKC;
ALTER TABLE ORDER_LINE33 ADD CONSTRAINT
ORDER_LINE33CKC CHECK (OL_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR ORDER_LINE33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE34 OFF;
ALTER TABLE ORDER_LINE34 DROP CONSTRAINT
ORDER_LINE34CKC;

```

```

ALTER TABLE ORDER_LINE34 ADD CONSTRAINT
ORDER_LINE34CKC CHECK (OL_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR ORDER_LINE34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE35 OFF;
ALTER TABLE ORDER_LINE35 DROP CONSTRAINT
ORDER_LINE35CKC;
ALTER TABLE ORDER_LINE35 ADD CONSTRAINT
ORDER_LINE35CKC CHECK (OL_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR ORDER_LINE35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE36 OFF;
ALTER TABLE ORDER_LINE36 DROP CONSTRAINT
ORDER_LINE36CKC;
ALTER TABLE ORDER_LINE36 ADD CONSTRAINT
ORDER_LINE36CKC CHECK (OL_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR ORDER_LINE36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE37 OFF;
ALTER TABLE ORDER_LINE37 DROP CONSTRAINT
ORDER_LINE37CKC;
ALTER TABLE ORDER_LINE37 ADD CONSTRAINT
ORDER_LINE37CKC CHECK (OL_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR ORDER_LINE37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE38 OFF;
ALTER TABLE ORDER_LINE38 DROP CONSTRAINT
ORDER_LINE38CKC;
ALTER TABLE ORDER_LINE38 ADD CONSTRAINT
ORDER_LINE38CKC CHECK (OL_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR ORDER_LINE38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE39 OFF;
ALTER TABLE ORDER_LINE39 DROP CONSTRAINT
ORDER_LINE39CKC;
ALTER TABLE ORDER_LINE39 ADD CONSTRAINT
ORDER_LINE39CKC CHECK (OL_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR ORDER_LINE39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE40 OFF;
ALTER TABLE ORDER_LINE40 DROP CONSTRAINT
ORDER_LINE40CKC;
ALTER TABLE ORDER_LINE40 ADD CONSTRAINT
ORDER_LINE40CKC CHECK (OL_W_ID >= 62401);
SET INTEGRITY FOR ORDER_LINE40 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST STOCK.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK1 OFF;
ALTER TABLE STOCK1 DROP CONSTRAINT STOCK1CKC;

```

```
ALTER TABLE STOCK1 ADD CONSTRAINT STOCK1CKC CHECK
(S_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK2 OFF;
ALTER TABLE STOCK2 DROP CONSTRAINT STOCK2CKC;
ALTER TABLE STOCK2 ADD CONSTRAINT STOCK2CKC CHECK
(S_W_ID BETWEEN 1601 AND 3200);
SET INTEGRITY FOR STOCK2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK3 OFF;
ALTER TABLE STOCK3 DROP CONSTRAINT STOCK3CKC;
ALTER TABLE STOCK3 ADD CONSTRAINT STOCK3CKC CHECK
(S_W_ID BETWEEN 3201 AND 4800);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK4 OFF;
ALTER TABLE STOCK4 DROP CONSTRAINT STOCK4CKC;
ALTER TABLE STOCK4 ADD CONSTRAINT STOCK4CKC CHECK
(S_W_ID BETWEEN 4801 AND 6400);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK5 OFF;
ALTER TABLE STOCK5 DROP CONSTRAINT STOCK5CKC;
ALTER TABLE STOCK5 ADD CONSTRAINT STOCK5CKC CHECK
(S_W_ID BETWEEN 6401 AND 8000);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE UNCHECKED;
```

```
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK6 OFF;
ALTER TABLE STOCK6 DROP CONSTRAINT STOCK6CKC;
ALTER TABLE STOCK6 ADD CONSTRAINT STOCK6CKC CHECK
(S_W_ID BETWEEN 8001 AND 9600);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK7 OFF;
ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC CHECK
(S_W_ID BETWEEN 9601 AND 11200);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK8 OFF;
ALTER TABLE STOCK8 DROP CONSTRAINT STOCK8CKC;
ALTER TABLE STOCK8 ADD CONSTRAINT STOCK8CKC CHECK
(S_W_ID BETWEEN 11201 AND 12800);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK9 OFF;
ALTER TABLE STOCK9 DROP CONSTRAINT STOCK9CKC;
ALTER TABLE STOCK9 ADD CONSTRAINT STOCK9CKC CHECK
(S_W_ID BETWEEN 12801 AND 14400);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK10 OFF;
ALTER TABLE STOCK10 DROP CONSTRAINT STOCK10CKC;
ALTER TABLE STOCK10 ADD CONSTRAINT STOCK10CKC
CHECK (S_W_ID BETWEEN 14401 AND 16000);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK11 OFF;
ALTER TABLE STOCK11 DROP CONSTRAINT STOCK11CKC;
```

```
ALTER TABLE STOCK11 ADD CONSTRAINT STOCK11CKC
CHECK (S_W_ID BETWEEN 16001 AND 17600);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK12 OFF;
ALTER TABLE STOCK12 DROP CONSTRAINT STOCK12CKC;
ALTER TABLE STOCK12 ADD CONSTRAINT STOCK12CKC
CHECK (S_W_ID BETWEEN 17601 AND 19200);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK13 OFF;
ALTER TABLE STOCK13 DROP CONSTRAINT STOCK13CKC;
ALTER TABLE STOCK13 ADD CONSTRAINT STOCK13CKC
CHECK (S_W_ID BETWEEN 19201 AND 20800);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK14 OFF;
ALTER TABLE STOCK14 DROP CONSTRAINT STOCK14CKC;
ALTER TABLE STOCK14 ADD CONSTRAINT STOCK14CKC
CHECK (S_W_ID BETWEEN 20801 AND 22400);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK15 OFF;
ALTER TABLE STOCK15 DROP CONSTRAINT STOCK15CKC;
ALTER TABLE STOCK15 ADD CONSTRAINT STOCK15CKC
CHECK (S_W_ID BETWEEN 22401 AND 24000);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK16 OFF;
ALTER TABLE STOCK16 DROP CONSTRAINT STOCK16CKC;
ALTER TABLE STOCK16 ADD CONSTRAINT STOCK16CKC
CHECK (S_W_ID BETWEEN 24001 AND 25600);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK17 OFF;
ALTER TABLE STOCK17 DROP CONSTRAINT STOCK17CKC;
ALTER TABLE STOCK17 ADD CONSTRAINT STOCK17CKC
CHECK (S_W_ID BETWEEN 25601 AND 27200);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK18 OFF;
ALTER TABLE STOCK18 DROP CONSTRAINT STOCK18CKC;
ALTER TABLE STOCK18 ADD CONSTRAINT STOCK18CKC
CHECK (S_W_ID BETWEEN 27201 AND 28800);
SET INTEGRITY FOR STOCK18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK19 OFF;
ALTER TABLE STOCK19 DROP CONSTRAINT STOCK19CKC;
ALTER TABLE STOCK19 ADD CONSTRAINT STOCK19CKC
CHECK (S_W_ID BETWEEN 28801 AND 30400);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK20 OFF;
ALTER TABLE STOCK20 DROP CONSTRAINT STOCK20CKC;
ALTER TABLE STOCK20 ADD CONSTRAINT STOCK20CKC
CHECK (S_W_ID BETWEEN 30401 AND 32000);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK21 OFF;
ALTER TABLE STOCK21 DROP CONSTRAINT STOCK21CKC;
ALTER TABLE STOCK21 ADD CONSTRAINT STOCK21CKC
CHECK (S_W_ID BETWEEN 32001 AND 33600);
```

```

SET INTEGRITY FOR STOCK21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK22 OFF;
ALTER TABLE STOCK22 DROP CONSTRAINT STOCK22CKC;
ALTER TABLE STOCK22 ADD CONSTRAINT STOCK22CKC
CHECK (S_W_ID BETWEEN 33601 AND 35200);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK23 OFF;
ALTER TABLE STOCK23 DROP CONSTRAINT STOCK23CKC;
ALTER TABLE STOCK23 ADD CONSTRAINT STOCK23CKC
CHECK (S_W_ID BETWEEN 35201 AND 36800);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK24 OFF;
ALTER TABLE STOCK24 DROP CONSTRAINT STOCK24CKC;
ALTER TABLE STOCK24 ADD CONSTRAINT STOCK24CKC
CHECK (S_W_ID BETWEEN 36801 AND 38400);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK25 OFF;
ALTER TABLE STOCK25 DROP CONSTRAINT STOCK25CKC;
ALTER TABLE STOCK25 ADD CONSTRAINT STOCK25CKC
CHECK (S_W_ID BETWEEN 38401 AND 40000);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK26 OFF;
ALTER TABLE STOCK26 DROP CONSTRAINT STOCK26CKC;
ALTER TABLE STOCK26 ADD CONSTRAINT STOCK26CKC
CHECK (S_W_ID BETWEEN 40001 AND 41600);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK27 OFF;
ALTER TABLE STOCK27 DROP CONSTRAINT STOCK27CKC;
ALTER TABLE STOCK27 ADD CONSTRAINT STOCK27CKC
CHECK (S_W_ID BETWEEN 41601 AND 43200);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK28 OFF;
ALTER TABLE STOCK28 DROP CONSTRAINT STOCK28CKC;
ALTER TABLE STOCK28 ADD CONSTRAINT STOCK28CKC
CHECK (S_W_ID BETWEEN 43201 AND 44800);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK29 OFF;
ALTER TABLE STOCK29 DROP CONSTRAINT STOCK29CKC;
ALTER TABLE STOCK29 ADD CONSTRAINT STOCK29CKC
CHECK (S_W_ID BETWEEN 44801 AND 46400);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK30 OFF;
ALTER TABLE STOCK30 DROP CONSTRAINT STOCK30CKC;
ALTER TABLE STOCK30 ADD CONSTRAINT STOCK30CKC
CHECK (S_W_ID BETWEEN 46401 AND 48000);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK31 OFF;
ALTER TABLE STOCK31 DROP CONSTRAINT STOCK31CKC;
ALTER TABLE STOCK31 ADD CONSTRAINT STOCK31CKC
CHECK (S_W_ID BETWEEN 48001 AND 49600);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK32 OFF;
ALTER TABLE STOCK32 DROP CONSTRAINT STOCK32CKC;
ALTER TABLE STOCK32 ADD CONSTRAINT STOCK32CKC
CHECK (S_W_ID BETWEEN 49601 AND 51200);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT STOCK33CKC
CHECK (S_W_ID BETWEEN 51201 AND 52800);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT STOCK34CKC
CHECK (S_W_ID BETWEEN 52801 AND 54400);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT STOCK35CKC
CHECK (S_W_ID BETWEEN 54401 AND 56000);
SET INTEGRITY FOR STOCK35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK36 OFF;
ALTER TABLE STOCK36 DROP CONSTRAINT STOCK36CKC;
ALTER TABLE STOCK36 ADD CONSTRAINT STOCK36CKC
CHECK (S_W_ID BETWEEN 56001 AND 57600);
SET INTEGRITY FOR STOCK36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK37 OFF;
ALTER TABLE STOCK37 DROP CONSTRAINT STOCK37CKC;
ALTER TABLE STOCK37 ADD CONSTRAINT STOCK37CKC
CHECK (S_W_ID BETWEEN 57601 AND 59200);
SET INTEGRITY FOR STOCK37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK38 OFF;
ALTER TABLE STOCK38 DROP CONSTRAINT STOCK38CKC;
ALTER TABLE STOCK38 ADD CONSTRAINT STOCK38CKC
CHECK (S_W_ID BETWEEN 59201 AND 60800);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK39 OFF;
ALTER TABLE STOCK39 DROP CONSTRAINT STOCK39CKC;
ALTER TABLE STOCK39 ADD CONSTRAINT STOCK39CKC
CHECK (S_W_ID BETWEEN 60801 AND 62400);
SET INTEGRITY FOR STOCK39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK40 OFF;
ALTER TABLE STOCK40 DROP CONSTRAINT STOCK40CKC;
ALTER TABLE STOCK40 ADD CONSTRAINT STOCK40CKC
CHECK (S_W_ID >= 62401);
SET INTEGRITY FOR STOCK40 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 8000);
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 8001 AND 16000);
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 16001 AND
24000);
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 24001 AND
32000);
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 32001 AND
40000);
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 40001 AND
48000);
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 48001 AND
56000);
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 >= 56001);

```

```

SET INTEGRITY FOR WAREHOUSE8 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;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB39;
CREATE INDEX CUST_IDXB39
      ON CUSTOMER39(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB40;
CREATE INDEX CUST_IDXB40
      ON CUSTOMER40(C_LAST, C_W_ID, C_D_ID,
C_FIRST, C_ID) PCTFREE 0;
connect reset;

```

CRIDX 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;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB39;
CREATE INDEX ORDR_IDXB39
    ON ORDERS39(O_C_ID, O_W_ID, O_D_ID, O_ID
DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB40;
CREATE INDEX ORDR_IDXB40
    ON ORDERS40(O_C_ID, O_W_ID, O_D_ID, O_ID
DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

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 1600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER2;
CREATE TABLE CUSTOMER2
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_02
INDEX IN is_customer_02
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 1601 ENDING AT 3200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER3;
CREATE TABLE CUSTOMER3

```

```

(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_03
INDEX IN is_customer_03
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 3201 ENDING AT 4800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER4;
CREATE TABLE CUSTOMER4
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_04
INDEX IN is_customer_04
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 4801 ENDING AT 6400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER5;
CREATE TABLE CUSTOMER5
(

```

```

C_ID      INTEGER    NOT NULL,
C_STATE   CHAR(2)    NOT NULL,
C_ZIP     CHAR(9)    NOT NULL,
C_PHONE   CHAR(16)   NOT NULL,
C_SINCE   BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT  NOT NULL,
C_MIDDLE  CHAR(2)    NOT NULL,
C_CREDIT  CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER  NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16) NOT NULL,
C_FIRST   VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT  NOT NULL,
C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT  NOT NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_05
INDEX IN is_customer_05
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 6401 ENDING AT 8000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER6;
CREATE TABLE CUSTOMER6
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT  NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_06
INDEX IN is_customer_06
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 8001 ENDING AT 9600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER7;
CREATE TABLE CUSTOMER7
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,

```

```

C_ZIP     CHAR(9)    NOT NULL,
C_PHONE   CHAR(16)   NOT NULL,
C_SINCE   BIGINT     NOT NULL,
C_CREDIT_LIM BIGINT  NOT NULL,
C_MIDDLE  CHAR(2)    NOT NULL,
C_CREDIT  CHAR(2)    NOT NULL,
C_DISCOUNT INTEGER  NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16) NOT NULL,
C_FIRST   VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT  NOT NULL,
C_W_ID    INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT  NOT NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_07
INDEX IN is_customer_07
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 9601 ENDING AT 11200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER8;
CREATE TABLE CUSTOMER8
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT  NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_08
INDEX IN is_customer_08
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 11201 ENDING AT 12800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER9;
CREATE TABLE CUSTOMER9
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,

```

```

C_SINCE      BIGINT    NOT NULL,
C_CREDIT_LIM BIGINT    NOT NULL,
C_MIDDLE     CHAR(2)   NOT NULL,
C_CREDIT     CHAR(2)   NOT NULL,
C_DISCOUNT  INTEGER   NOT NULL,
C_DATA       VARCHAR(500) NOT NULL,
C_LAST       VARCHAR(16) NOT NULL,
C_FIRST      VARCHAR(16) NOT NULL,

C_STREET_1   VARCHAR(20) NOT NULL,
C_STREET_2   VARCHAR(20) NOT NULL,
C_CITY       VARCHAR(20) NOT NULL,
C_D_ID       SMALLINT  NOT NULL,
C_W_ID       INTEGER   NOT NULL,
C_DELIVERY_CNT INTEGER   NOT NULL,
C_BALANCE    BIGINT    NOT NULL,
C_YTD_PAYMENT BIGINT    NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_09
INDEX IN is_customer_09
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 12801 ENDING AT 14400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER10;
CREATE TABLE CUSTOMER10
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER   NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_10
INDEX IN is_customer_10
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 14401 ENDING AT 16000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER11;
CREATE TABLE CUSTOMER11
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,

```

```

C_CREDIT_LIM BIGINT    NOT NULL,
C_MIDDLE     CHAR(2)   NOT NULL,
C_CREDIT     CHAR(2)   NOT NULL,
C_DISCOUNT  INTEGER   NOT NULL,
C_DATA       VARCHAR(500) NOT NULL,
C_LAST       VARCHAR(16) NOT NULL,
C_FIRST      VARCHAR(16) NOT NULL,
C_STREET_1   VARCHAR(20) NOT NULL,
C_STREET_2   VARCHAR(20) NOT NULL,
C_CITY       VARCHAR(20) NOT NULL,
C_D_ID       SMALLINT  NOT NULL,
C_W_ID       INTEGER   NOT NULL,
C_DELIVERY_CNT INTEGER   NOT NULL,
C_BALANCE    BIGINT    NOT NULL,
C_YTD_PAYMENT BIGINT    NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_11
INDEX IN is_customer_11
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 16001 ENDING AT 17600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER12;
CREATE TABLE CUSTOMER12
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,

  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER   NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_12
INDEX IN is_customer_12
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 17601 ENDING AT 19200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER13;
CREATE TABLE CUSTOMER13
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_13
INDEX IN is_customer_13
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 19201 ENDING AT 20800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER14;
CREATE TABLE CUSTOMER14

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_14
INDEX IN is_customer_14
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 20801 ENDING AT 22400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER15;
CREATE TABLE CUSTOMER15

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,

C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,

```

```

C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_15
INDEX IN is_customer_15
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 22401 ENDING AT 24000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER16;
CREATE TABLE CUSTOMER16

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_16
INDEX IN is_customer_16
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 24001 ENDING AT 25600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER17;
CREATE TABLE CUSTOMER17

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,

```

```

C_DATA    VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16)  NOT NULL,
C_FIRST  VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT   NOT NULL,
C_W_ID   INTEGER     NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_17
INDEX IN is_customer_17
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 25601 ENDING AT 27200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER18;
CREATE TABLE CUSTOMER18
(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)     NOT NULL,
  C_ZIP     CHAR(9)     NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_18
INDEX IN is_customer_18
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 27201 ENDING AT 28800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER19;
CREATE TABLE CUSTOMER19
(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)     NOT NULL,
  C_ZIP     CHAR(9)     NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,

```

```

C_DATA    VARCHAR(500) NOT NULL,
C_LAST   VARCHAR(16)  NOT NULL,
C_FIRST  VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY   VARCHAR(20) NOT NULL,
C_D_ID   SMALLINT   NOT NULL,
C_W_ID   INTEGER     NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_19
INDEX IN is_customer_19
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 28801 ENDING AT 30400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER20;
CREATE TABLE CUSTOMER20
(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)     NOT NULL,
  C_ZIP     CHAR(9)     NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_20
INDEX IN is_customer_20
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 30401 ENDING AT 32000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER21;
CREATE TABLE CUSTOMER21
(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)     NOT NULL,
  C_ZIP     CHAR(9)     NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,

```

```

C_FIRST    VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY     VARCHAR(20) NOT NULL,
C_D_ID     SMALLINT  NOT NULL,
C_W_ID     INTEGER   NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE  BIGINT   NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_21
INDEX IN is_customer_21
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 32001 ENDING AT 33600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER22;
CREATE TABLE CUSTOMER22
(
  C_ID      INTEGER   NOT NULL,
  C_STATE   CHAR(2)   NOT NULL,
  C_ZIP     CHAR(9)   NOT NULL,
  C_PHONE   CHAR(16)  NOT NULL,
  C_SINCE   BIGINT   NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE  CHAR(2)   NOT NULL,
  C_CREDIT  CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER   NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE  BIGINT   NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_22
INDEX IN is_customer_22
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 33601 ENDING AT 35200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER23;
CREATE TABLE CUSTOMER23
(
  C_ID      INTEGER   NOT NULL,
  C_STATE   CHAR(2)   NOT NULL,
  C_ZIP     CHAR(9)   NOT NULL,
  C_PHONE   CHAR(16)  NOT NULL,
  C_SINCE   BIGINT   NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE  CHAR(2)   NOT NULL,
  C_CREDIT  CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,

```

```

C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY     VARCHAR(20) NOT NULL,
C_D_ID     SMALLINT  NOT NULL,
C_W_ID     INTEGER   NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE  BIGINT   NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_23
INDEX IN is_customer_23
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 35201 ENDING AT 36800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER24;
CREATE TABLE CUSTOMER24
(
  C_ID      INTEGER   NOT NULL,
  C_STATE   CHAR(2)   NOT NULL,
  C_ZIP     CHAR(9)   NOT NULL,
  C_PHONE   CHAR(16)  NOT NULL,
  C_SINCE   BIGINT   NOT NULL,
  C_CREDIT_LIM BIGINT NOT NULL,
  C_MIDDLE  CHAR(2)   NOT NULL,
  C_CREDIT  CHAR(2)   NOT NULL,
  C_DISCOUNT INTEGER NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER   NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE  BIGINT   NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_24
INDEX IN is_customer_24
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 36801 ENDING AT 38400,
  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 38401 ENDING AT 40000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER26;
CREATE TABLE CUSTOMER26

```

```

(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_26
INDEX IN is_customer_26
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 40001 ENDING AT 41600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER27;
CREATE TABLE CUSTOMER27

```

```

(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER     NOT NULL,

```

```

C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_27
INDEX IN is_customer_27
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 41601 ENDING AT 43200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER28;
CREATE TABLE CUSTOMER28

```

```

(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_28
INDEX IN is_customer_28
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 43201 ENDING AT 44800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER29;
CREATE TABLE CUSTOMER29

```

```

(
  C_ID      INTEGER     NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT  NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT INTEGER  NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT  NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,

```



```

C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_29
INDEX IN is_customer_29
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 44801 ENDING AT 46400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER30;
CREATE TABLE CUSTOMER30

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_30
INDEX IN is_customer_30
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 46401 ENDING AT 48000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER31;
CREATE TABLE CUSTOMER31

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

)
IN ts_customer_31
INDEX IN is_customer_31
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 48001 ENDING AT 49600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER32;
CREATE TABLE CUSTOMER32

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_32
INDEX IN is_customer_32
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 49601 ENDING AT 51200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_33

INDEX IN is_customer_33
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 51201 ENDING AT 52800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_34
INDEX IN is_customer_34
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 52801 ENDING AT 54400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_35

```

```

INDEX IN is_customer_35
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 54401 ENDING AT 56000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_36
INDEX IN is_customer_36
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 56001 ENDING AT 57600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER37;
CREATE TABLE CUSTOMER37
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_37
INDEX IN is_customer_37
ORGANIZE BY KEY SEQUENCE (

```

```

C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 57601 ENDING AT 59200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER38;
CREATE TABLE CUSTOMER38
(
C_ID      INTEGER      NOT NULL,
C_STATE   CHAR(2)      NOT NULL,
C_ZIP     CHAR(9)      NOT NULL,
C_PHONE   CHAR(16)     NOT NULL,
C_SINCE   BIGINT      NOT NULL,
C_CREDIT_LIM BIGINT    NOT NULL,
C_MIDDLE  CHAR(2)     NOT NULL,
C_CREDIT  CHAR(2)     NOT NULL,
C_DISCOUNT INTEGER    NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16)  NOT NULL,
C_FIRST   VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20)  NOT NULL,
C_D_ID    SMALLINT    NOT NULL,
C_W_ID    INTEGER     NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE BIGINT      NOT NULL,
C_YTD_PAYMENT BIGINT   NOT NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_38
INDEX IN is_customer_38
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 59201 ENDING AT 60800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER39;
CREATE TABLE CUSTOMER39
(
C_ID      INTEGER      NOT NULL,
C_STATE   CHAR(2)      NOT NULL,
C_ZIP     CHAR(9)      NOT NULL,
C_PHONE   CHAR(16)     NOT NULL,
C_SINCE   BIGINT      NOT NULL,
C_CREDIT_LIM BIGINT    NOT NULL,
C_MIDDLE  CHAR(2)     NOT NULL,
C_CREDIT  CHAR(2)     NOT NULL,
C_DISCOUNT INTEGER    NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16)  NOT NULL,
C_FIRST   VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20)  NOT NULL,
C_D_ID    SMALLINT    NOT NULL,
C_W_ID    INTEGER     NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE BIGINT      NOT NULL,
C_YTD_PAYMENT BIGINT   NOT NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_39
INDEX IN is_customer_39
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 60801 ENDING AT 62400,

```

```

C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER40;
CREATE TABLE CUSTOMER40
(
C_ID      INTEGER      NOT NULL,
C_STATE   CHAR(2)      NOT NULL,
C_ZIP     CHAR(9)      NOT NULL,
C_PHONE   CHAR(16)     NOT NULL,
C_SINCE   BIGINT      NOT NULL,
C_CREDIT_LIM BIGINT    NOT NULL,
C_MIDDLE  CHAR(2)     NOT NULL,
C_CREDIT  CHAR(2)     NOT NULL,
C_DISCOUNT INTEGER    NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16)  NOT NULL,
C_FIRST   VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20)  NOT NULL,
C_D_ID    SMALLINT    NOT NULL,
C_W_ID    INTEGER     NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE BIGINT      NOT NULL,
C_YTD_PAYMENT BIGINT   NOT NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_40
INDEX IN is_customer_40
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 62401 ENDING AT 64000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;

```

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 8000
)
)
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 8001 ENDING AT 16000
)
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 16001 ENDING AT 24000
)
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 24001 ENDING AT 32000
)
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 32001 ENDING AT 40000
)
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 40001 ENDING AT 48000
)
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 48001 ENDING AT 56000
)
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 56001 ENDING AT 64000
)
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_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_08
INDEX IN ts_history_08;
ALTER TABLE HISTORY8 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 100000
)
ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;

```

CRTB_NEW_ORDER.ddl

```

connect to TPCC in share mode;

DROP TABLE NEW_ORDERA1;
CREATE TABLE NEW_ORDERA1
(
  NO_O_ID     INTEGER NOT NULL,
  NO_D_ID     SMALLINT NOT NULL,
  NO_W_ID     INTEGER NOT NULL
)
IN ts_newordA_01
INDEX IN ts_newordA_01
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 1 ENDING AT 8000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA2;
CREATE TABLE NEW_ORDERA2
(
  NO_O_ID     INTEGER NOT NULL,

```

```

  NO_D_ID     SMALLINT NOT NULL,
  NO_W_ID     INTEGER NOT NULL
)
IN ts_newordA_02
INDEX IN ts_newordA_02
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 8001 ENDING AT 16000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA3;
CREATE TABLE NEW_ORDERA3
(
  NO_O_ID     INTEGER NOT NULL,
  NO_D_ID     SMALLINT NOT NULL,
  NO_W_ID     INTEGER NOT NULL
)
IN ts_newordA_03
INDEX IN ts_newordA_03
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 16001 ENDING AT
24000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA4;
CREATE TABLE NEW_ORDERA4
(
  NO_O_ID     INTEGER NOT NULL,
  NO_D_ID     SMALLINT NOT NULL,
  NO_W_ID     INTEGER NOT NULL
)
IN ts_newordA_04
INDEX IN ts_newordA_04
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 24001 ENDING AT
32000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA5;
CREATE TABLE NEW_ORDERA5
(
  NO_O_ID     INTEGER NOT NULL,
  NO_D_ID     SMALLINT NOT NULL,
  NO_W_ID     INTEGER NOT NULL
)
IN ts_newordA_05
INDEX IN ts_newordA_05
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 32001 ENDING AT
40000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA6;
CREATE TABLE NEW_ORDERA6
(
  NO_O_ID     INTEGER NOT NULL,
  NO_D_ID     SMALLINT NOT NULL,

```

```

NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_06
INDEX IN ts_newordA_06
ORGANIZE BY KEY SEQUENCE (
48000,    NO_W_ID STARTING FROM 40001 ENDING AT
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA7;
CREATE TABLE NEW_ORDERA7
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_07
INDEX IN ts_newordA_07
ORGANIZE BY KEY SEQUENCE (
56000,    NO_W_ID STARTING FROM 48001 ENDING AT
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA8;
CREATE TABLE NEW_ORDERA8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_08
INDEX IN ts_newordA_08
ORGANIZE BY KEY SEQUENCE (
64000,    NO_W_ID STARTING FROM 56001 ENDING AT
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 1900 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB1;
CREATE TABLE NEW_ORDERB1
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_01
INDEX IN ts_newordB_01
ORGANIZE BY KEY SEQUENCE (
          NO_W_ID STARTING FROM 1 ENDING AT 8000,
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB2;
CREATE TABLE NEW_ORDERB2
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL

```

```

)
IN ts_newordB_02
INDEX IN ts_newordB_02
ORGANIZE BY KEY SEQUENCE (
          NO_W_ID STARTING FROM 8001 ENDING AT 16000,
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB3;
CREATE TABLE NEW_ORDERB3
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_03
INDEX IN ts_newordB_03
ORGANIZE BY KEY SEQUENCE (
24000,    NO_W_ID STARTING FROM 16001 ENDING AT
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB4;
CREATE TABLE NEW_ORDERB4
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_04
INDEX IN ts_newordB_04
ORGANIZE BY KEY SEQUENCE (
32000,    NO_W_ID STARTING FROM 24001 ENDING AT
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB5;
CREATE TABLE NEW_ORDERB5
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_05
INDEX IN ts_newordB_05
ORGANIZE BY KEY SEQUENCE (
40000,    NO_W_ID STARTING FROM 32001 ENDING AT
          NO_D_ID STARTING FROM 1 ENDING AT 10,
          NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB6;
CREATE TABLE NEW_ORDERB6
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)

```

```

IN ts_newordB_06
INDEX IN ts_newordB_06
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 40001 ENDING AT
48000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB7;
CREATE TABLE NEW_ORDERB7
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_07
INDEX IN ts_newordB_07
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 48001 ENDING AT
56000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB8;
CREATE TABLE NEW_ORDERB8
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_08
INDEX IN ts_newordB_08
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 56001 ENDING AT
64000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5451
)
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 0 ENDING AT 3675,
O_W_ID STARTING FROM 1 ENDING AT 1600,
O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS2;
CREATE TABLE ORDERS2
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_02
INDEX IN is_order_02
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 1601 ENDING AT 3200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS3;
CREATE TABLE ORDERS3
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_03
INDEX IN is_order_03
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 3201 ENDING AT 4800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS4;
CREATE TABLE ORDERS4
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_04
INDEX IN is_order_04
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 4801 ENDING AT 6400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS5;
CREATE TABLE ORDERS5

```



```

(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_05
INDEX IN is_order_05
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 6401 ENDING AT 8000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS6;
CREATE TABLE ORDERS6
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_06
INDEX IN is_order_06
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 8001 ENDING AT 9600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS7;
CREATE TABLE ORDERS7
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_07
INDEX IN is_order_07
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 9601 ENDING AT 11200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS8;
CREATE TABLE ORDERS8
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,

```

```

O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_08
INDEX IN is_order_08
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 11201 ENDING AT 12800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS9;
CREATE TABLE ORDERS9
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_09
INDEX IN is_order_09
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 12801 ENDING AT 14400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS10;
CREATE TABLE ORDERS10
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_10
INDEX IN is_order_10
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 14401 ENDING AT 16000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS11;
CREATE TABLE ORDERS11
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_11
INDEX IN is_order_11

```

```

ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 16001 ENDING AT 17600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS12;
CREATE TABLE ORDERS12
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT    NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER    NOT NULL,
  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_12
INDEX IN is_order_12
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 17601 ENDING AT 19200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS13;
CREATE TABLE ORDERS13
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT    NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER    NOT NULL,
  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_13
INDEX IN is_order_13
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 19201 ENDING AT 20800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS14;
CREATE TABLE ORDERS14
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT    NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER    NOT NULL,
  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_14
INDEX IN is_order_14
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 20801 ENDING AT 22400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS15;
CREATE TABLE ORDERS15
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT    NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER    NOT NULL,
  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_15
INDEX IN is_order_15
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 22401 ENDING AT 24000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS16;
CREATE TABLE ORDERS16
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT    NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER    NOT NULL,
  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_16
INDEX IN is_order_16
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 24001 ENDING AT 25600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS17;
CREATE TABLE ORDERS17
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT    NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER    NOT NULL,
  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_17
INDEX IN is_order_17
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 25601 ENDING AT 27200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS18;
CREATE TABLE ORDERS18
(
  O_C_ID    INTEGER    NOT NULL,

```

```

O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_18
INDEX IN is_order_18
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 27201 ENDING AT 28800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS19;
CREATE TABLE ORDERS19
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_19
INDEX IN is_order_19
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 28801 ENDING AT 30400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS20;
CREATE TABLE ORDERS20
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_20
INDEX IN is_order_20
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 30401 ENDING AT 32000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS21;
CREATE TABLE ORDERS21
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,

```

```

O_D_ID SMALLINT NOT NULL
)
IN ts_order_21
INDEX IN is_order_21
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 32001 ENDING AT 33600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS22;
CREATE TABLE ORDERS22
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_22
INDEX IN is_order_22
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 33601 ENDING AT 35200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS23;
CREATE TABLE ORDERS23
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_23
INDEX IN is_order_23
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 35201 ENDING AT 36800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS24;
CREATE TABLE ORDERS24
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,

O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_24
INDEX IN is_order_24
ORGANIZE BY KEY SEQUENCE (

```

```

O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 36801 ENDING AT 38400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS25;
CREATE TABLE ORDERS25
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_25
INDEX IN is_order_25
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 38401 ENDING AT 40000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS26;
CREATE TABLE ORDERS26
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_26
INDEX IN is_order_26
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 40001 ENDING AT 41600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS27;
CREATE TABLE ORDERS27
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_27
INDEX IN is_order_27
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 41601 ENDING AT 43200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS28;
CREATE TABLE ORDERS28
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_28
INDEX IN is_order_28
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 43201 ENDING AT 44800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS29;
CREATE TABLE ORDERS29
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_29
INDEX IN is_order_29
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 44801 ENDING AT 46400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS30;
CREATE TABLE ORDERS30
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_30
INDEX IN is_order_30
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 46401 ENDING AT 48000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS31;
CREATE TABLE ORDERS31
(
O_C_ID INTEGER NOT NULL,

```

```

O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_31
INDEX IN is_order_31
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 48001 ENDING AT 49600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS32;
CREATE TABLE ORDERS32
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_32
INDEX IN is_order_32
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 49601 ENDING AT 51200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS33;
CREATE TABLE ORDERS33
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_33
INDEX IN is_order_33
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 51201 ENDING AT 52800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS34;
CREATE TABLE ORDERS34
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,

```

```

O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_34
INDEX IN is_order_34
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 52801 ENDING AT 54400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS35;
CREATE TABLE ORDERS35
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_35
INDEX IN is_order_35
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 54401 ENDING AT 56000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS36;
CREATE TABLE ORDERS36
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_36
INDEX IN is_order_36
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3675,
O_W_ID STARTING FROM 56001 ENDING AT 57600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS37;
CREATE TABLE ORDERS37
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_37
INDEX IN is_order_37

```

```

ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 57601 ENDING AT 59200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS38;
CREATE TABLE ORDERS38
(
  O_C_ID      INTEGER NOT NULL,
  O_ENTRY_D   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID        INTEGER NOT NULL,
  O_W_ID      INTEGER NOT NULL,
  O_D_ID      SMALLINT NOT NULL
)
IN ts_order_38
INDEX IN is_order_38
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 59201 ENDING AT 60800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS39;
CREATE TABLE ORDERS39
(
  O_C_ID      INTEGER NOT NULL,
  O_ENTRY_D   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID        INTEGER NOT NULL,
  O_W_ID      INTEGER NOT NULL,
  O_D_ID      SMALLINT NOT NULL
)
IN ts_order_39
INDEX IN is_order_39
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 60801 ENDING AT 62400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS40;
CREATE TABLE ORDERS40
(
  O_C_ID      INTEGER NOT NULL,
  O_ENTRY_D   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID        INTEGER NOT NULL,
  O_W_ID      INTEGER NOT NULL,
  O_D_ID      SMALLINT NOT NULL
)
IN ts_order_40
INDEX IN is_order_40
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3675,
  O_W_ID STARTING FROM 62401 ENDING AT 64000,
  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 1600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 1601 ENDING AT 3200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 3201 ENDING AT 4800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 4801 ENDING AT 6400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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_LINES5
(
  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 6401 ENDING AT 8000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 8001 ENDING AT 9600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 9601 ENDING AT 11200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 11201 ENDING AT
12800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 12801 ENDING AT
14400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 14401 ENDING AT
16000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 16001 ENDING AT
17600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 17601 ENDING AT
19200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 19201 ENDING AT
20800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 (

```



```

22400, OL_W_ID STARTING FROM 20801 ENDING AT
      OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 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 22401 ENDING AT

```

```

24000, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 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 24001 ENDING AT

```

```

25600, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 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 25601 ENDING AT

```

```

27200, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 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 27201 ENDING AT

```

```

28800, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 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 28801 ENDING AT

```

```

30400, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 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 30401 ENDING AT
32000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 32001 ENDING AT
33600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 33601 ENDING AT
35200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 35201 ENDING AT
36800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 36801 ENDING AT
38400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 38401 ENDING AT
40000,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 40001 ENDING AT
41600,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 41601 ENDING AT
43200,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 43201 ENDING AT
44800,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 44801 ENDING AT
46400,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 46401 ENDING AT
48000,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 48001 ENDING AT
49600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 49601 ENDING AT
51200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 51201 ENDING AT
52800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 52801 ENDING AT
54400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 54401 ENDING AT
56000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 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 (

```

```

57600, OL_W_ID STARTING FROM 56001 ENDING AT
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 57601 ENDING AT

```

```

59200, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 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 59201 ENDING AT

```

```

60800, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE39;
CREATE TABLE ORDER_LINE39

```

```

(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,

```

```

OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_39
INDEX IN ts_orderline_39
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 60801 ENDING AT

```

```

62400, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE40;
CREATE TABLE ORDER_LINE40

```

```

(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_40
INDEX IN ts_orderline_40
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 62401 ENDING AT

```

```

64000, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 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 1600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK2;
CREATE TABLE STOCK2
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_02
INDEX IN ts_stock_02
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 1601 ENDING AT 3200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK3;
CREATE TABLE STOCK3
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_03
INDEX IN ts_stock_03
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 3201 ENDING AT 4800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK4;
CREATE TABLE STOCK4
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,

```

```

  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_04
INDEX IN ts_stock_04
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 4801 ENDING AT 6400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK5;
CREATE TABLE STOCK5
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_05
INDEX IN ts_stock_05
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 6401 ENDING AT 8000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,

```

```

S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_06
INDEX IN ts_stock_06
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 8001 ENDING AT 9600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK7;
CREATE TABLE STOCK7
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_07
INDEX IN ts_stock_07
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 9601 ENDING AT 11200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK8;
CREATE TABLE STOCK8
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_08
INDEX IN ts_stock_08
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 11201 ENDING AT 12800
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE STOCK9;
CREATE TABLE STOCK9
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_09
INDEX IN ts_stock_09
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 12801 ENDING AT 14400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK10;
CREATE TABLE STOCK10
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_10
INDEX IN ts_stock_10
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 14401 ENDING AT 16000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK11;
CREATE TABLE STOCK11
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,

```

```

S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_11
INDEX IN ts_stock_11
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 16001 ENDING AT 17600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK12;
CREATE TABLE STOCK12
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_12
INDEX IN ts_stock_12
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 17601 ENDING AT 19200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK13;
CREATE TABLE STOCK13
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_13
INDEX IN ts_stock_13

```

```

ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 19201 ENDING AT 20800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK14;
CREATE TABLE STOCK14
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_14
INDEX IN ts_stock_14
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 20801 ENDING AT 22400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK15;
CREATE TABLE STOCK15
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_15
INDEX IN ts_stock_15
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 22401 ENDING AT 24000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK16;
CREATE TABLE STOCK16
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,

```



```

S_ORDER_CNT INTEGER NOT NULL,
S_YTD      INTEGER NOT NULL,
S_DATA     VARCHAR(50) NOT NULL,
S_DIST_01  CHAR(24) NOT NULL,
S_DIST_02  CHAR(24) NOT NULL,
S_DIST_03  CHAR(24) NOT NULL,
S_DIST_04  CHAR(24) NOT NULL,
S_DIST_05  CHAR(24) NOT NULL,
S_DIST_06  CHAR(24) NOT NULL,
S_DIST_07  CHAR(24) NOT NULL,
S_DIST_08  CHAR(24) NOT NULL,
S_DIST_09  CHAR(24) NOT NULL,
S_DIST_10  CHAR(24) NOT NULL,
S_I_ID     INTEGER NOT NULL,
S_W_ID     INTEGER NOT NULL
)
IN ts_stock_16
INDEX IN ts_stock_16
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 24001 ENDING AT 25600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK17;
CREATE TABLE STOCK17
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24) NOT NULL,
S_DIST_02   CHAR(24) NOT NULL,
S_DIST_03   CHAR(24) NOT NULL,
S_DIST_04   CHAR(24) NOT NULL,
S_DIST_05   CHAR(24) NOT NULL,
S_DIST_06   CHAR(24) NOT NULL,
S_DIST_07   CHAR(24) NOT NULL,
S_DIST_08   CHAR(24) NOT NULL,
S_DIST_09   CHAR(24) NOT NULL,
S_DIST_10   CHAR(24) NOT NULL,
S_I_ID      INTEGER NOT NULL,
S_W_ID      INTEGER NOT NULL
)
IN ts_stock_17
INDEX IN ts_stock_17
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 25601 ENDING AT 27200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK18;
CREATE TABLE STOCK18
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24) NOT NULL,
S_DIST_02   CHAR(24) NOT NULL,
S_DIST_03   CHAR(24) NOT NULL,
S_DIST_04   CHAR(24) NOT NULL,
S_DIST_05   CHAR(24) NOT NULL,
S_DIST_06   CHAR(24) NOT NULL,
S_DIST_07   CHAR(24) NOT NULL,
S_DIST_08   CHAR(24) NOT NULL,
S_DIST_09   CHAR(24) NOT NULL,

```

```

S_DIST_10   CHAR(24) NOT NULL,
S_I_ID      INTEGER NOT NULL,
S_W_ID      INTEGER NOT NULL
)
IN ts_stock_18
INDEX IN ts_stock_18
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 27201 ENDING AT 28800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK19;
CREATE TABLE STOCK19
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24) NOT NULL,
S_DIST_02   CHAR(24) NOT NULL,
S_DIST_03   CHAR(24) NOT NULL,
S_DIST_04   CHAR(24) NOT NULL,
S_DIST_05   CHAR(24) NOT NULL,
S_DIST_06   CHAR(24) NOT NULL,
S_DIST_07   CHAR(24) NOT NULL,
S_DIST_08   CHAR(24) NOT NULL,
S_DIST_09   CHAR(24) NOT NULL,
S_DIST_10   CHAR(24) NOT NULL,
S_I_ID      INTEGER NOT NULL,
S_W_ID      INTEGER NOT NULL
)
IN ts_stock_19
INDEX IN ts_stock_19
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 28801 ENDING AT 30400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK20;
CREATE TABLE STOCK20
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24) NOT NULL,
S_DIST_02   CHAR(24) NOT NULL,
S_DIST_03   CHAR(24) NOT NULL,
S_DIST_04   CHAR(24) NOT NULL,
S_DIST_05   CHAR(24) NOT NULL,
S_DIST_06   CHAR(24) NOT NULL,
S_DIST_07   CHAR(24) NOT NULL,
S_DIST_08   CHAR(24) NOT NULL,
S_DIST_09   CHAR(24) NOT NULL,
S_DIST_10   CHAR(24) NOT NULL,
S_I_ID      INTEGER NOT NULL,
S_W_ID      INTEGER NOT NULL
)
IN ts_stock_20
INDEX IN ts_stock_20
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 30401 ENDING AT 32000
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK21;
CREATE TABLE STOCK21
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_21
INDEX IN ts_stock_21
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 32001 ENDING AT 33600
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK22;
CREATE TABLE STOCK22
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,

  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_22
INDEX IN ts_stock_22
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 33601 ENDING AT 35200
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK23;
CREATE TABLE STOCK23
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,

```

```

  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_23
INDEX IN ts_stock_23
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 35201 ENDING AT 36800
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK24;
CREATE TABLE STOCK24
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_24
INDEX IN ts_stock_24
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 36801 ENDING AT 38400
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK25;
CREATE TABLE STOCK25
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,

  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

)
IN ts_stock_25
INDEX IN ts_stock_25
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 38401 ENDING AT 40000
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK26;
CREATE TABLE STOCK26

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_26
INDEX IN ts_stock_26
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 40001 ENDING AT 41600
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK27;
CREATE TABLE STOCK27

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_27
INDEX IN ts_stock_27
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 41601 ENDING AT 43200
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK28;
CREATE TABLE STOCK28

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,

```

```

  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_28
INDEX IN ts_stock_28
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 43201 ENDING AT 44800
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK29;
CREATE TABLE STOCK29

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_29
INDEX IN ts_stock_29
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 44801 ENDING AT 46400
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK30;
CREATE TABLE STOCK30

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,

```

```

S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_30
INDEX IN ts_stock_30
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 46401 ENDING AT 48000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK31;
CREATE TABLE STOCK31
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_31
INDEX IN ts_stock_31
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 48001 ENDING AT 49600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK32;
CREATE TABLE STOCK32
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_32
INDEX IN ts_stock_32
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,

```

```

S_W_ID STARTING FROM 49601 ENDING AT 51200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_33
INDEX IN ts_stock_33
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 51201 ENDING AT 52800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK34;
CREATE TABLE STOCK34
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_34
INDEX IN ts_stock_34
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 52801 ENDING AT 54400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK35;
CREATE TABLE STOCK35
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_35
INDEX IN ts_stock_35
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 54401 ENDING AT 56000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK36;
CREATE TABLE STOCK36
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_36
INDEX IN ts_stock_36
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 56001 ENDING AT 57600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK37;
CREATE TABLE STOCK37
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,

```

```

S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_37
INDEX IN ts_stock_37
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 57601 ENDING AT 59200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK38;
CREATE TABLE STOCK38
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_38
INDEX IN ts_stock_38
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 59201 ENDING AT 60800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_39
INDEX IN ts_stock_39
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 60801 ENDING AT 62400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK40;

```

```

CREATE TABLE STOCK40
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
IN ts_stock_40
INDEX IN ts_stock_40
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 62401 ENDING AT 64000
)
ALLOW OVERFLOW;
connect reset;

```

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 8000
)
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 8001 ENDING AT 16000
)
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 16001 ENDING AT 24000
)
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 24001 ENDING AT 32000
)
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 32001 ENDING AT 40000
)
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 40001 ENDING AT 48000
)
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 48001 ENDING AT 56000
)
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 56001 ENDING AT 64000
)
ALLOW OVERFLOW;
connect reset;

```

CRVW CUSTOMER.ddl

```

connect to TPCC in share mode;

DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER
(C_ID,
C_STATE,
C_ZIP,
C_PHONE,

```

```

C_SINCE,
C_CREDIT_LIM,
C_MIDDLE,
C_CREDIT,
C_DISCOUNT,
C_DATA,
C_LAST,
C_FIRST,
C_STREET_1,
C_STREET_2,
C_CITY,
C_D_ID,
C_W_ID,
C_DELIVERY_CNT,
C_BALANCE,
C_YTD_PAYMENT,
C_PAYMENT_CNT
) AS SELECT * FROM CUSTOMER1 UNION ALL
SELECT * FROM CUSTOMER2 UNION ALL
SELECT * FROM CUSTOMER3 UNION ALL
SELECT * FROM CUSTOMER4 UNION ALL
SELECT * FROM CUSTOMER5 UNION ALL
SELECT * FROM CUSTOMER6 UNION ALL
SELECT * FROM CUSTOMER7 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL
SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW DISTRICT.ddl

```

connect to TPCC in share mode;

DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
(D_NEXT_O_ID,
D_TAX,

```

```

D_YTD,
D_NAME,
D_STREET_1,
D_STREET_2,
D_CITY,
D_STATE,
D_ZIP,
D_ID,
D_W_ID
) AS SELECT * FROM DISTRICT1 UNION ALL
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL
SELECT * FROM DISTRICT8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW HISTORY.ddl

connect to TPCC in share mode;

```

DROP VIEW HISTORY;
CREATE VIEW HISTORY
(H_C_ID,
H_C_D_ID,
H_C_W_ID,
H_D_ID,
H_W_ID,
H_DATE,
H_AMOUNT,
H_DATA
) AS SELECT * FROM HISTORY1 UNION ALL
SELECT * FROM HISTORY2 UNION ALL
SELECT * FROM HISTORY3 UNION ALL
SELECT * FROM HISTORY4 UNION ALL
SELECT * FROM HISTORY5 UNION ALL
SELECT * FROM HISTORY6 UNION ALL
SELECT * FROM HISTORY7 UNION ALL
SELECT * FROM HISTORY8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW NEW ORDER.ddl

connect to TPCC in share mode;

```

DROP VIEW NEW_ORDER;
CREATE VIEW NEW_ORDER
(NO_O_ID,
NO_D_ID,
NO_W_ID
) AS SELECT * FROM NEW_ORDERA1 UNION ALL
SELECT * FROM NEW_ORDERA2 UNION ALL
SELECT * FROM NEW_ORDERA3 UNION ALL
SELECT * FROM NEW_ORDERA4 UNION ALL
SELECT * FROM NEW_ORDERA5 UNION ALL
SELECT * FROM NEW_ORDERA6 UNION ALL
SELECT * FROM NEW_ORDERA7 UNION ALL
SELECT * FROM NEW_ORDERA8 UNION ALL
SELECT * FROM NEW_ORDERB1 UNION ALL
SELECT * FROM NEW_ORDERB2 UNION ALL
SELECT * FROM NEW_ORDERB3 UNION ALL

```

```

SELECT * FROM NEW_ORDERB4 UNION ALL
SELECT * FROM NEW_ORDERB5 UNION ALL
SELECT * FROM NEW_ORDERB6 UNION ALL
SELECT * FROM NEW_ORDERB7 UNION ALL
SELECT * FROM NEW_ORDERB8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW ORDERS.ddl

connect to TPCC in share mode;

```

DROP VIEW ORDERS;
CREATE VIEW ORDERS
(O_C_ID,
O_ENTRY_D,
O_CARRIER_ID,
O_OL_CNT,
O_ALL_LOCAL,
O_ID,
O_W_ID,
O_D_ID
) AS SELECT * FROM ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION ALL
SELECT * FROM ORDERS3 UNION ALL
SELECT * FROM ORDERS4 UNION ALL
SELECT * FROM ORDERS5 UNION ALL
SELECT * FROM ORDERS6 UNION ALL
SELECT * FROM ORDERS7 UNION ALL
SELECT * FROM ORDERS8 UNION ALL
SELECT * FROM ORDERS9 UNION ALL
SELECT * FROM ORDERS10 UNION ALL
SELECT * FROM ORDERS11 UNION ALL
SELECT * FROM ORDERS12 UNION ALL
SELECT * FROM ORDERS13 UNION ALL
SELECT * FROM ORDERS14 UNION ALL
SELECT * FROM ORDERS15 UNION ALL
SELECT * FROM ORDERS16 UNION ALL
SELECT * FROM ORDERS17 UNION ALL
SELECT * FROM ORDERS18 UNION ALL
SELECT * FROM ORDERS19 UNION ALL
SELECT * FROM ORDERS20 UNION ALL
SELECT * FROM ORDERS21 UNION ALL
SELECT * FROM ORDERS22 UNION ALL
SELECT * FROM ORDERS23 UNION ALL
SELECT * FROM ORDERS24 UNION ALL
SELECT * FROM ORDERS25 UNION ALL
SELECT * FROM ORDERS26 UNION ALL
SELECT * FROM ORDERS27 UNION ALL
SELECT * FROM ORDERS28 UNION ALL
SELECT * FROM ORDERS29 UNION ALL
SELECT * FROM ORDERS30 UNION ALL
SELECT * FROM ORDERS31 UNION ALL
SELECT * FROM ORDERS32 UNION ALL
SELECT * FROM ORDERS33 UNION ALL
SELECT * FROM ORDERS34 UNION ALL
SELECT * FROM ORDERS35 UNION ALL
SELECT * FROM ORDERS36 UNION ALL
SELECT * FROM ORDERS37 UNION ALL
SELECT * FROM ORDERS38 UNION ALL
SELECT * FROM ORDERS39 UNION ALL
SELECT * FROM ORDERS40
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 UNION ALL
SELECT * FROM ORDER_LINE39 UNION ALL
SELECT * FROM ORDER_LINE40
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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

CRVW 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 8000
)
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 8001 ENDING AT 16000
)
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 16001 ENDING AT 24000
)
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 24001 ENDING AT 32000
)
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 32001 ENDING AT 40000
)
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 40001 ENDING AT 48000
)
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 48001 ENDING AT 56000
)
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 56001 ENDING AT 64000
)
ALLOW OVERFLOW;

```

connect reset;

GEN_CUSTOMER.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 400 -f1
/flats/F1_01/customer_01_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 401 800 -f1
/flats/F1_01/customer_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 801 1200 -f1
/flats/F1_01/customer_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1201 1600 -f1
/flats/F1_01/customer_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1601 2000 -f1
/flats/F1_02/customer_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2001 2400 -f1
/flats/F1_02/customer_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2401 2800 -f1
/flats/F1_02/customer_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2801 3200 -f1
/flats/F1_02/customer_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3201 3600 -f1
/flats/F1_03/customer_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3601 4000 -f1
/flats/F1_03/customer_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4001 4400 -f1
/flats/F1_03/customer_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4401 4800 -f1
/flats/F1_03/customer_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4801 5200 -f1
/flats/F1_04/customer_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5201 5600 -f1
/flats/F1_04/customer_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5601 6000 -f1
/flats/F1_04/customer_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6001 6400 -f1
/flats/F1_04/customer_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6401 6800 -f1
/flats/F1_05/customer_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6801 7200 -f1
/flats/F1_05/customer_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7201 7600 -f1
/flats/F1_05/customer_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7601 8000 -f1
/flats/F1_05/customer_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8001 8400 -f1
/flats/F1_06/customer_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8401 8800 -f1
/flats/F1_06/customer_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8801 9200 -f1
/flats/F1_06/customer_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9201 9600 -f1
/flats/F1_06/customer_06_4.dat

```

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9601 10000 -f1
/flats/F1_07/customer_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10001 10400 -f1
/flats/F1_07/customer_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10401 10800 -f1
/flats/F1_07/customer_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10801 11200 -f1
/flats/F1_07/customer_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11201 11600 -f1
/flats/F1_08/customer_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11601 12000 -f1
/flats/F1_08/customer_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12001 12400 -f1
/flats/F1_08/customer_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12401 12800 -f1
/flats/F1_08/customer_08_4.dat

```

GEN_DISTRICT.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 8000 -f1
/flats/F1_01/district_01_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 8001 16000 -f1
/flats/F1_02/district_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 16001 24000 -f1
/flats/F1_03/district_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 24001 32000 -f1
/flats/F1_04/district_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 32001 40000 -f1
/flats/F1_05/district_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 40001 48000 -f1
/flats/F1_06/district_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 48001 56000 -f1
/flats/F1_07/district_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 56001 64000 -f1
/flats/F1_08/district_08_1.dat

```

GEN_HISTORY.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 8000 -f1
/flats/F1_01/history_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 8001 16000 -f1
/flats/F1_02/history_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 16001 24000 -f1
/flats/F1_03/history_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 24001 32000 -f1
/flats/F1_04/history_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 32001 40000 -f1
/flats/F1_05/history_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 40001 48000 -f1
/flats/F1_06/history_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 48001 56000 -f1
/flats/F1_07/history_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 56001 64000 -f1
/flats/F1_08/history_08_1.dat

```

GEN_ITEM.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 5 -f1 /flats/F1_01/item_1.dat

```


GEN STOCK.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 400 -f1
/flats/F1_01/stock_01_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 401 800 -f1
/flats/F1_01/stock_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 801 1200 -f1
/flats/F1_01/stock_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1201 1600 -f1
/flats/F1_01/stock_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1601 2000 -f1
/flats/F1_02/stock_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2001 2400 -f1
/flats/F1_02/stock_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2401 2800 -f1
/flats/F1_02/stock_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2801 3200 -f1
/flats/F1_02/stock_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3201 3600 -f1
/flats/F1_03/stock_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3601 4000 -f1
/flats/F1_03/stock_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4001 4400 -f1
/flats/F1_03/stock_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4401 4800 -f1
/flats/F1_03/stock_03_4.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4801 5200 -f1
/flats/F1_04/stock_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5201 5600 -f1
/flats/F1_04/stock_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5601 6000 -f1
/flats/F1_04/stock_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6001 6400 -f1
/flats/F1_04/stock_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6401 6800 -f1
/flats/F1_05/stock_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6801 7200 -f1
/flats/F1_05/stock_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7201 7600 -f1
/flats/F1_05/stock_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7601 8000 -f1
/flats/F1_05/stock_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8001 8400 -f1
/flats/F1_06/stock_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8401 8800 -f1
/flats/F1_06/stock_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8801 9200 -f1
/flats/F1_06/stock_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9201 9600 -f1
/flats/F1_06/stock_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9601 10000 -f1
/flats/F1_07/stock_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10001 10400 -f1
/flats/F1_07/stock_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10401 10800 -f1
/flats/F1_07/stock_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10801 11200 -f1
/flats/F1_07/stock_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11201 11600 -f1
/flats/F1_08/stock_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11601 12000 -f1
/flats/F1_08/stock_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12001 12400 -f1
/flats/F1_08/stock_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12401 12800 -f1
/flats/F1_08/stock_08_4.dat
```

GEN WAREHOUSE.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1 8000 -f1
/flats/F1_01/warehouse_01_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 8001 16000 -f1
/flats/F1_02/warehouse_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 16001 24000 -f1
/flats/F1_03/warehouse_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 24001 32000 -f1
/flats/F1_04/warehouse_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 32001 40000 -f1
/flats/F1_05/warehouse_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 40001 48000 -f1
/flats/F1_06/warehouse_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 48001 56000 -f1
/flats/F1_07/warehouse_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 56001 64000 -f1
/flats/F1_08/warehouse_08_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/F1_01/customer_01_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 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/F1_01/customer_01_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 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/F1_01/customer_01_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 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/F1_01/customer_01_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_02/customer_02_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
```



```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_06/customer_06_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_06/customer_06_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_06/customer_06_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_07/customer_07_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_07/customer_07_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_07/customer_07_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_07/customer_07_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_08/customer_08_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_08/customer_08_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_08/customer_08_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_08/customer_08_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;

```

LOAD DISTRICT ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;

IMPORT FROM /flats/F1_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/F1_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/F1_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/F1_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/F1_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/F1_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/F1_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/F1_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;
```

LOAD HISTORY ALL.ddl

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_01/history_01_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY1 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_02/history_02_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY2 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_03/history_03_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY3 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_04/history_04_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY4 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_05/history_05_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY5 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_06/history_06_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY6 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_07/history_07_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY7 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

connect to TPCC in share mode;

```
LOAD FROM /flats/F1_08/history_08_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY8 NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
```

connect reset;

LOAD ITEM 1.ddl

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_01/item_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 1000 INSERT INTO ITEM; COMMIT WORK; CONNECT RESET;
```

LOAD NEW ORDER ALL.ddl

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_01/neworder_01_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_01/neworder_01_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_01/neworder_01_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_01/neworder_01_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_02/neworder_02_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_02/neworder_02_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_02/neworder_02_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2; COMMIT WORK; CONNECT RESET;
```

CONNECT TO TPCC IN SHARE MODE;

```
IMPORT FROM /flats/F1_02/neworder_02_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2; COMMIT WORK; CONNECT RESET;
```



```

WRITE ACCESS COMMITCOUNT 20000 INSERT INTO
NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_08/neworder_08_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 20000 INSERT INTO
NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_08/neworder_08_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 20000 INSERT INTO
NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_08/neworder_08_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 20000 INSERT INTO
NEW_ORDERA8;
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/F1_01/orders_01_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_01/orders_01_2.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_01/orders_01_3.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_01/orders_01_4.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_02/orders_02_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_02/orders_02_2.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_02/orders_02_3.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_02/orders_02_4.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_03/orders_03_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_03/orders_03_2.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_03/orders_03_3.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_03/orders_03_4.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_04/orders_04_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 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/F1_04/orders_04_2.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
12000000 INSERT INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;

```



```
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_08/orderline_08_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
132000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
```

```
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_01/stock_01_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_01/stock_01_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_01/stock_01_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_01/stock_01_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_02/stock_02_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_02/stock_02_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_02/stock_02_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
```

```
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_02/stock_02_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_03/stock_03_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_03/stock_03_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_03/stock_03_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_03/stock_03_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_04/stock_04_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_04/stock_04_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_04/stock_04_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_04/stock_04_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
```

```

ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_05/stock_05_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_05/stock_05_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_05/stock_05_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_05/stock_05_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_06/stock_06_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_06/stock_06_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_06/stock_06_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_06/stock_06_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_07/stock_07_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;

```

```

UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_07/stock_07_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_07/stock_07_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_07/stock_07_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_08/stock_08_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_08/stock_08_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_08/stock_08_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/F1_08/stock_08_4.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
40000000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;

```

LOAD WAREHOUSE ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;

IMPORT FROM /flats/F1_01/warehouse_01_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_02/warehouse_02_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW

```

```

WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_03/warehouse_03_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_04/warehouse_04_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_05/warehouse_05_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_06/warehouse_06_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_07/warehouse_07_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_08/warehouse_08_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50 ALLOW
WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE8;
COMMIT WORK;
CONNECT RESET;

```

RNST_CUSTOMER.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER5 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER8 AND INDEXES ALL;
COMMIT WORK;
connect 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;
RUNSTATS ON TABLE TPCC.CUSTOMER16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER23 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```


connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.CUSTOMER30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.DISTRICT1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_HISTORY.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.HISTORY1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.HISTORY8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ITEM.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.ITEM AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDER.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.NEW_ORDERA1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERA8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.NEW_ORDERB3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDERB8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.ORDERS1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS9 AND INDEXES ALL;
COMMIT WORK;
connect reset;

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE TPCC.STOCK1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK8 AND INDEXES ALL;
COMMIT WORK;

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK26 AND INDEXES ALL;
COMMIT WORK;

```

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

RNST WAREHOUSE.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE1 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE2 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

```

```

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE3 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE4 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE5 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE6 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE7 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE8 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

```

SORT ORDERS.sh

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_01/orders_01_1.dat >
/flats/F1_01/orders_01_1.dat.sorted

if [ $? = 0 ]
then
mv /flats/F1_01/orders_01_1.dat.sorted
/flats/F1_01/orders_01_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_01/orders_01_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_01/orders_01_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

echo "**** WARNING ****"
echo "Sort of /flats/F1_01/orders_01_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_01/orders_01_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_01/orders_01_4.dat >
/flats/F1_01/orders_01_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_01/orders_01_4.dat.sorted
/flats/F1_01/orders_01_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_01/orders_01_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_01/orders_01_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_02/orders_02_1.dat >
/flats/F1_02/orders_02_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_02/orders_02_1.dat.sorted
/flats/F1_02/orders_02_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_02/orders_02_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_02/orders_02_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_02/orders_02_2.dat >
/flats/F1_02/orders_02_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_02/orders_02_2.dat.sorted
/flats/F1_02/orders_02_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_02/orders_02_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_02/orders_02_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_02/orders_02_3.dat >
/flats/F1_02/orders_02_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_02/orders_02_3.dat.sorted
/flats/F1_02/orders_02_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_02/orders_02_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_02/orders_02_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_02/orders_02_4.dat >
/flats/F1_02/orders_02_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_02/orders_02_4.dat.sorted
/flats/F1_02/orders_02_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_02/orders_02_4.dat FAILED."

```

```

echo "Please ensure that the source file
(/flats/F1_02/orders_02_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_03/orders_03_1.dat >
/flats/F1_03/orders_03_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_03/orders_03_1.dat.sorted
/flats/F1_03/orders_03_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_03/orders_03_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_03/orders_03_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_03/orders_03_2.dat >
/flats/F1_03/orders_03_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_03/orders_03_2.dat.sorted
/flats/F1_03/orders_03_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_03/orders_03_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_03/orders_03_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_03/orders_03_3.dat >
/flats/F1_03/orders_03_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_03/orders_03_3.dat.sorted
/flats/F1_03/orders_03_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_03/orders_03_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_03/orders_03_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_03/orders_03_4.dat >
/flats/F1_03/orders_03_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_03/orders_03_4.dat.sorted
/flats/F1_03/orders_03_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_03/orders_03_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_03/orders_03_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_04/orders_04_1.dat >
/flats/F1_04/orders_04_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_04/orders_04_1.dat.sorted
/flats/F1_04/orders_04_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_04/orders_04_1.dat FAILED."

```

```

echo "Please ensure that the source file
(/flats/F1_04/orders_04_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_04/orders_04_2.dat >
/flats/F1_04/orders_04_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_04/orders_04_2.dat.sorted
/flats/F1_04/orders_04_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_04/orders_04_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_04/orders_04_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_04/orders_04_3.dat >
/flats/F1_04/orders_04_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_04/orders_04_3.dat.sorted
/flats/F1_04/orders_04_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_04/orders_04_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_04/orders_04_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_04/orders_04_4.dat >
/flats/F1_04/orders_04_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_04/orders_04_4.dat.sorted
/flats/F1_04/orders_04_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_04/orders_04_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_04/orders_04_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_05/orders_05_1.dat >
/flats/F1_05/orders_05_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/orders_05_1.dat.sorted
/flats/F1_05/orders_05_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/orders_05_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/orders_05_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_05/orders_05_2.dat >
/flats/F1_05/orders_05_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/orders_05_2.dat.sorted
/flats/F1_05/orders_05_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/orders_05_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/orders_05_2.dat)"

```

```

echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_05/orders_05_3.dat >
/flats/F1_05/orders_05_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/orders_05_3.dat.sorted
/flats/F1_05/orders_05_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/orders_05_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/orders_05_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_05/orders_05_4.dat >
/flats/F1_05/orders_05_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/orders_05_4.dat.sorted
/flats/F1_05/orders_05_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/orders_05_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/orders_05_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_06/orders_06_1.dat >
/flats/F1_06/orders_06_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/orders_06_1.dat.sorted
/flats/F1_06/orders_06_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/orders_06_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/orders_06_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_06/orders_06_2.dat >
/flats/F1_06/orders_06_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/orders_06_2.dat.sorted
/flats/F1_06/orders_06_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/orders_06_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/orders_06_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_06/orders_06_3.dat >
/flats/F1_06/orders_06_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/orders_06_3.dat.sorted
/flats/F1_06/orders_06_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/orders_06_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/orders_06_3.dat)"

```



```

echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_06/orders_06_4.dat >
/flats/F1_06/orders_06_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/orders_06_4.dat.sorted
/flats/F1_06/orders_06_4.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_06/orders_06_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/orders_06_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_07/orders_07_1.dat >
/flats/F1_07/orders_07_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/orders_07_1.dat.sorted
/flats/F1_07/orders_07_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_07/orders_07_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/orders_07_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_07/orders_07_2.dat >
/flats/F1_07/orders_07_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/orders_07_2.dat.sorted
/flats/F1_07/orders_07_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_07/orders_07_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/orders_07_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_07/orders_07_3.dat >
/flats/F1_07/orders_07_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/orders_07_3.dat.sorted
/flats/F1_07/orders_07_3.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_07/orders_07_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/orders_07_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_07/orders_07_4.dat >
/flats/F1_07/orders_07_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/orders_07_4.dat.sorted
/flats/F1_07/orders_07_4.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_07/orders_07_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/orders_07_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_08/orders_08_1.dat >
/flats/F1_08/orders_08_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/orders_08_1.dat.sorted
/flats/F1_08/orders_08_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_08/orders_08_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/orders_08_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_08/orders_08_2.dat >
/flats/F1_08/orders_08_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/orders_08_2.dat.sorted
/flats/F1_08/orders_08_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_08/orders_08_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/orders_08_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_08/orders_08_3.dat >
/flats/F1_08/orders_08_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/orders_08_3.dat.sorted
/flats/F1_08/orders_08_3.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_08/orders_08_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/orders_08_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_08/orders_08_4.dat >
/flats/F1_08/orders_08_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/orders_08_4.dat.sorted
/flats/F1_08/orders_08_4.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_08/orders_08_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/orders_08_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_09/orders_09_1.dat >
/flats/F1_09/orders_09_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_09/orders_09_1.dat.sorted
/flats/F1_09/orders_09_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/F1_09/orders_09_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_09/orders_09_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_09/orders_09_2.dat >
/flats/F1_09/orders_09_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_09/orders_09_2.dat.sorted
/flats/F1_09/orders_09_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_09/orders_09_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_09/orders_09_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_09/orders_09_3.dat >
/flats/F1_09/orders_09_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_09/orders_09_3.dat.sorted
/flats/F1_09/orders_09_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_09/orders_09_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_09/orders_09_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_09/orders_09_4.dat >
/flats/F1_09/orders_09_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_09/orders_09_4.dat.sorted
/flats/F1_09/orders_09_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_09/orders_09_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_09/orders_09_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_10/orders_10_1.dat >
/flats/F1_10/orders_10_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_10/orders_10_1.dat.sorted
/flats/F1_10/orders_10_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_10/orders_10_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_10/orders_10_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_10/orders_10_2.dat >
/flats/F1_10/orders_10_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_10/orders_10_2.dat.sorted
/flats/F1_10/orders_10_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_10/orders_10_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_10/orders_10_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_10/orders_10_3.dat >
/flats/F1_10/orders_10_3.dat.sorted

```

```

if [ $? = 0 ]
then
mv /flats/F1_10/orders_10_3.dat.sorted
/flats/F1_10/orders_10_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_10/orders_10_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_10/orders_10_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_10/orders_10_4.dat >
/flats/F1_10/orders_10_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_10/orders_10_4.dat.sorted
/flats/F1_10/orders_10_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_10/orders_10_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_10/orders_10_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_11/orders_11_1.dat >
/flats/F1_11/orders_11_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_11/orders_11_1.dat.sorted
/flats/F1_11/orders_11_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_11/orders_11_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_11/orders_11_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_11/orders_11_2.dat >
/flats/F1_11/orders_11_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_11/orders_11_2.dat.sorted
/flats/F1_11/orders_11_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_11/orders_11_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_11/orders_11_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_11/orders_11_3.dat >
/flats/F1_11/orders_11_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_11/orders_11_3.dat.sorted
/flats/F1_11/orders_11_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_11/orders_11_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_11/orders_11_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_11/orders_11_4.dat >
/flats/F1_11/orders_11_4.dat.sorted
if [ $? = 0 ]
then

```

```

mv /flats/F1_11/orders_11_4.dat.sorted
/flats/F1_11/orders_11_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_11/orders_11_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_11/orders_11_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_12/orders_12_1.dat >
/flats/F1_12/orders_12_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_12/orders_12_1.dat.sorted
/flats/F1_12/orders_12_1.dat
else
echo "**** WARNING ****"

echo "Sort of /flats/F1_12/orders_12_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_12/orders_12_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_12/orders_12_2.dat >
/flats/F1_12/orders_12_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_12/orders_12_2.dat.sorted
/flats/F1_12/orders_12_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_12/orders_12_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_12/orders_12_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_12/orders_12_3.dat >
/flats/F1_12/orders_12_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_12/orders_12_3.dat.sorted
/flats/F1_12/orders_12_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_12/orders_12_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_12/orders_12_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_12/orders_12_4.dat >
/flats/F1_12/orders_12_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_12/orders_12_4.dat.sorted
/flats/F1_12/orders_12_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_12/orders_12_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_12/orders_12_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_13/orders_13_1.dat >
/flats/F1_13/orders_13_1.dat.sorted
if [ $? = 0 ]
then

```

```

mv /flats/F1_13/orders_13_1.dat.sorted
/flats/F1_13/orders_13_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_13/orders_13_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_13/orders_13_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_13/orders_13_2.dat >
/flats/F1_13/orders_13_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_13/orders_13_2.dat.sorted
/flats/F1_13/orders_13_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_13/orders_13_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_13/orders_13_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_13/orders_13_3.dat >
/flats/F1_13/orders_13_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_13/orders_13_3.dat.sorted
/flats/F1_13/orders_13_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_13/orders_13_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_13/orders_13_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_13/orders_13_4.dat >
/flats/F1_13/orders_13_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_13/orders_13_4.dat.sorted
/flats/F1_13/orders_13_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_13/orders_13_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_13/orders_13_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_14/orders_14_1.dat >
/flats/F1_14/orders_14_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_14/orders_14_1.dat.sorted
/flats/F1_14/orders_14_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_14/orders_14_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_14/orders_14_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_14/orders_14_2.dat >
/flats/F1_14/orders_14_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_14/orders_14_2.dat.sorted
/flats/F1_14/orders_14_2.dat

```

```

else
echo "**** WARNING ****"
echo "Sort of /flats/F1_14/orders_14_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_14/orders_14_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_14/orders_14_3.dat >
/flats/F1_14/orders_14_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_14/orders_14_3.dat.sorted
/flats/F1_14/orders_14_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_14/orders_14_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_14/orders_14_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_14/orders_14_4.dat >
/flats/F1_14/orders_14_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_14/orders_14_4.dat.sorted
/flats/F1_14/orders_14_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_14/orders_14_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_14/orders_14_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_15/orders_15_1.dat >
/flats/F1_15/orders_15_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_15/orders_15_1.dat.sorted
/flats/F1_15/orders_15_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_15/orders_15_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_15/orders_15_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_15/orders_15_2.dat >
/flats/F1_15/orders_15_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_15/orders_15_2.dat.sorted
/flats/F1_15/orders_15_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_15/orders_15_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_15/orders_15_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_15/orders_15_3.dat >
/flats/F1_15/orders_15_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_15/orders_15_3.dat.sorted
/flats/F1_15/orders_15_3.dat
else
echo "**** WARNING ****"

```

```

echo "Sort of /flats/F1_15/orders_15_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_15/orders_15_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_15/orders_15_4.dat >
/flats/F1_15/orders_15_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_15/orders_15_4.dat.sorted
/flats/F1_15/orders_15_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_15/orders_15_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_15/orders_15_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_16/orders_16_1.dat >
/flats/F1_16/orders_16_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_16/orders_16_1.dat.sorted
/flats/F1_16/orders_16_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_16/orders_16_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_16/orders_16_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_16/orders_16_2.dat >
/flats/F1_16/orders_16_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_16/orders_16_2.dat.sorted
/flats/F1_16/orders_16_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_16/orders_16_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_16/orders_16_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_16/orders_16_3.dat >
/flats/F1_16/orders_16_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_16/orders_16_3.dat.sorted
/flats/F1_16/orders_16_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_16/orders_16_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_16/orders_16_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_16/orders_16_4.dat >
/flats/F1_16/orders_16_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_16/orders_16_4.dat.sorted
/flats/F1_16/orders_16_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_16/orders_16_4.dat FAILED."

```

```

echo "Please ensure that the source file
(/flats/F1_16/orders_16_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_17/orders_17_1.dat >
/flats/F1_17/orders_17_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_17/orders_17_1.dat.sorted
/flats/F1_17/orders_17_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_17/orders_17_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_17/orders_17_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_17/orders_17_2.dat >
/flats/F1_17/orders_17_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_17/orders_17_2.dat.sorted
/flats/F1_17/orders_17_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_17/orders_17_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_17/orders_17_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_17/orders_17_3.dat >
/flats/F1_17/orders_17_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_17/orders_17_3.dat.sorted
/flats/F1_17/orders_17_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_17/orders_17_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_17/orders_17_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_17/orders_17_4.dat >
/flats/F1_17/orders_17_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_17/orders_17_4.dat.sorted
/flats/F1_17/orders_17_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_17/orders_17_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_17/orders_17_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_18/orders_18_1.dat >
/flats/F1_18/orders_18_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_18/orders_18_1.dat.sorted
/flats/F1_18/orders_18_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_18/orders_18_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_18/orders_18_1.dat)"

```

```

echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_18/orders_18_2.dat >
/flats/F1_18/orders_18_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_18/orders_18_2.dat.sorted
/flats/F1_18/orders_18_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_18/orders_18_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_18/orders_18_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_18/orders_18_3.dat >
/flats/F1_18/orders_18_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_18/orders_18_3.dat.sorted
/flats/F1_18/orders_18_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_18/orders_18_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_18/orders_18_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_18/orders_18_4.dat >
/flats/F1_18/orders_18_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_18/orders_18_4.dat.sorted
/flats/F1_18/orders_18_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_18/orders_18_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_18/orders_18_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_19/orders_19_1.dat >
/flats/F1_19/orders_19_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_19/orders_19_1.dat.sorted
/flats/F1_19/orders_19_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_19/orders_19_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_19/orders_19_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_19/orders_19_2.dat >
/flats/F1_19/orders_19_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_19/orders_19_2.dat.sorted
/flats/F1_19/orders_19_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_19/orders_19_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_19/orders_19_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_19/orders_19_3.dat >
/flats/F1_19/orders_19_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_19/orders_19_3.dat.sorted
/flats/F1_19/orders_19_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_19/orders_19_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_19/orders_19_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_19/orders_19_4.dat >
/flats/F1_19/orders_19_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_19/orders_19_4.dat.sorted
/flats/F1_19/orders_19_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_19/orders_19_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_19/orders_19_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_20/orders_20_1.dat >
/flats/F1_20/orders_20_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_20/orders_20_1.dat.sorted
/flats/F1_20/orders_20_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_20/orders_20_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_20/orders_20_1.dat)"
echo "exists and that sufficient free space is available"

echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_20/orders_20_2.dat >
/flats/F1_20/orders_20_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_20/orders_20_2.dat.sorted
/flats/F1_20/orders_20_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_20/orders_20_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_20/orders_20_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_20/orders_20_3.dat >
/flats/F1_20/orders_20_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_20/orders_20_3.dat.sorted
/flats/F1_20/orders_20_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_20/orders_20_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_20/orders_20_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_20/orders_20_4.dat >
/flats/F1_20/orders_20_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_20/orders_20_4.dat.sorted
/flats/F1_20/orders_20_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_20/orders_20_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_20/orders_20_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_21/orders_21_1.dat >
/flats/F1_21/orders_21_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_21/orders_21_1.dat.sorted
/flats/F1_21/orders_21_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_21/orders_21_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_21/orders_21_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_21/orders_21_2.dat >
/flats/F1_21/orders_21_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_21/orders_21_2.dat.sorted
/flats/F1_21/orders_21_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_21/orders_21_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_21/orders_21_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_21/orders_21_3.dat >
/flats/F1_21/orders_21_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_21/orders_21_3.dat.sorted
/flats/F1_21/orders_21_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_21/orders_21_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_21/orders_21_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_21/orders_21_4.dat >
/flats/F1_21/orders_21_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_21/orders_21_4.dat.sorted
/flats/F1_21/orders_21_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_21/orders_21_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_21/orders_21_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_22/orders_22_1.dat >
/flats/F1_22/orders_22_1.dat.sorted

```

```

if [ $? = 0 ]
then
  mv /flats/F1_22/orders_22_1.dat.sorted
/flats/F1_22/orders_22_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_22/orders_22_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_22/orders_22_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_22/orders_22_2.dat >
/flats/F1_22/orders_22_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_22/orders_22_2.dat.sorted
/flats/F1_22/orders_22_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_22/orders_22_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_22/orders_22_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_22/orders_22_3.dat >
/flats/F1_22/orders_22_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_22/orders_22_3.dat.sorted
/flats/F1_22/orders_22_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_22/orders_22_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_22/orders_22_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_22/orders_22_4.dat >
/flats/F1_22/orders_22_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_22/orders_22_4.dat.sorted
/flats/F1_22/orders_22_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_22/orders_22_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_22/orders_22_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_23/orders_23_1.dat >
/flats/F1_23/orders_23_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_23/orders_23_1.dat.sorted
/flats/F1_23/orders_23_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_23/orders_23_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_23/orders_23_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_23/orders_23_2.dat >
/flats/F1_23/orders_23_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_23/orders_23_2.dat.sorted
/flats/F1_23/orders_23_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_23/orders_23_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_23/orders_23_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

  mv /flats/F1_23/orders_23_2.dat.sorted
/flats/F1_23/orders_23_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_23/orders_23_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_23/orders_23_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_23/orders_23_3.dat >
/flats/F1_23/orders_23_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_23/orders_23_3.dat.sorted
/flats/F1_23/orders_23_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_23/orders_23_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_23/orders_23_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_23/orders_23_4.dat >
/flats/F1_23/orders_23_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_23/orders_23_4.dat.sorted
/flats/F1_23/orders_23_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_23/orders_23_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_23/orders_23_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_24/orders_24_1.dat >
/flats/F1_24/orders_24_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_24/orders_24_1.dat.sorted
/flats/F1_24/orders_24_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_24/orders_24_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_24/orders_24_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_24/orders_24_2.dat >
/flats/F1_24/orders_24_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_24/orders_24_2.dat.sorted
/flats/F1_24/orders_24_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_24/orders_24_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_24/orders_24_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_24/orders_24_3.dat >
/flats/F1_24/orders_24_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_24/orders_24_3.dat.sorted
/flats/F1_24/orders_24_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_24/orders_24_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_24/orders_24_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_24/orders_24_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_24/orders_24_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_24/orders_24_4.dat >
/flats/F1_24/orders_24_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_24/orders_24_4.dat.sorted
/flats/F1_24/orders_24_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_24/orders_24_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_24/orders_24_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_25/orders_25_1.dat >
/flats/F1_25/orders_25_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_25/orders_25_1.dat.sorted
/flats/F1_25/orders_25_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_25/orders_25_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_25/orders_25_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_25/orders_25_2.dat >
/flats/F1_25/orders_25_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_25/orders_25_2.dat.sorted
/flats/F1_25/orders_25_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_25/orders_25_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_25/orders_25_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_25/orders_25_3.dat >
/flats/F1_25/orders_25_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_25/orders_25_3.dat.sorted
/flats/F1_25/orders_25_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_25/orders_25_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_25/orders_25_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_25/orders_25_4.dat >
/flats/F1_25/orders_25_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_25/orders_25_4.dat.sorted
/flats/F1_25/orders_25_4.dat
else
  echo "**** WARNING ****"

```

```

  echo "Sort of /flats/F1_25/orders_25_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_25/orders_25_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_26/orders_26_1.dat >
/flats/F1_26/orders_26_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_26/orders_26_1.dat.sorted
/flats/F1_26/orders_26_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_26/orders_26_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_26/orders_26_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_26/orders_26_2.dat >
/flats/F1_26/orders_26_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_26/orders_26_2.dat.sorted
/flats/F1_26/orders_26_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_26/orders_26_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_26/orders_26_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_26/orders_26_3.dat >
/flats/F1_26/orders_26_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_26/orders_26_3.dat.sorted
/flats/F1_26/orders_26_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_26/orders_26_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_26/orders_26_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_26/orders_26_4.dat >
/flats/F1_26/orders_26_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_26/orders_26_4.dat.sorted
/flats/F1_26/orders_26_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_26/orders_26_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_26/orders_26_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_27/orders_27_1.dat >
/flats/F1_27/orders_27_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_27/orders_27_1.dat.sorted
/flats/F1_27/orders_27_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_27/orders_27_1.dat FAILED."

```



```

echo "Please ensure that the source file
(/flats/F1_27/orders_27_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_27/orders_27_2.dat >
/flats/F1_27/orders_27_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_27/orders_27_2.dat.sorted
/flats/F1_27/orders_27_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_27/orders_27_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_27/orders_27_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_27/orders_27_3.dat >
/flats/F1_27/orders_27_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_27/orders_27_3.dat.sorted
/flats/F1_27/orders_27_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_27/orders_27_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_27/orders_27_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_27/orders_27_4.dat >
/flats/F1_27/orders_27_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_27/orders_27_4.dat.sorted
/flats/F1_27/orders_27_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_27/orders_27_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_27/orders_27_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_28/orders_28_1.dat >
/flats/F1_28/orders_28_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_28/orders_28_1.dat.sorted
/flats/F1_28/orders_28_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_28/orders_28_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_28/orders_28_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_28/orders_28_2.dat >
/flats/F1_28/orders_28_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_28/orders_28_2.dat.sorted
/flats/F1_28/orders_28_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_28/orders_28_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_28/orders_28_2.dat)"

```

```

echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_28/orders_28_3.dat >
/flats/F1_28/orders_28_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_28/orders_28_3.dat.sorted
/flats/F1_28/orders_28_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_28/orders_28_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_28/orders_28_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_28/orders_28_4.dat >
/flats/F1_28/orders_28_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_28/orders_28_4.dat.sorted
/flats/F1_28/orders_28_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_28/orders_28_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_28/orders_28_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_29/orders_29_1.dat >
/flats/F1_29/orders_29_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_29/orders_29_1.dat.sorted
/flats/F1_29/orders_29_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_29/orders_29_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_29/orders_29_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_29/orders_29_2.dat >
/flats/F1_29/orders_29_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_29/orders_29_2.dat.sorted
/flats/F1_29/orders_29_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_29/orders_29_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_29/orders_29_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_29/orders_29_3.dat >
/flats/F1_29/orders_29_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_29/orders_29_3.dat.sorted
/flats/F1_29/orders_29_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_29/orders_29_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_29/orders_29_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_29/orders_29_4.dat >
/flats/F1_29/orders_29_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_29/orders_29_4.dat.sorted
/flats/F1_29/orders_29_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_29/orders_29_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_29/orders_29_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_30/orders_30_1.dat >
/flats/F1_30/orders_30_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_30/orders_30_1.dat.sorted
/flats/F1_30/orders_30_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_30/orders_30_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_30/orders_30_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_30/orders_30_2.dat >
/flats/F1_30/orders_30_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_30/orders_30_2.dat.sorted
/flats/F1_30/orders_30_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_30/orders_30_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_30/orders_30_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_30/orders_30_3.dat >
/flats/F1_30/orders_30_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_30/orders_30_3.dat.sorted
/flats/F1_30/orders_30_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_30/orders_30_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_30/orders_30_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_30/orders_30_4.dat >
/flats/F1_30/orders_30_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_30/orders_30_4.dat.sorted
/flats/F1_30/orders_30_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_30/orders_30_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_30/orders_30_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_31/orders_31_1.dat >
/flats/F1_31/orders_31_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_31/orders_31_1.dat.sorted
/flats/F1_31/orders_31_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_31/orders_31_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_31/orders_31_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_31/orders_31_2.dat >
/flats/F1_31/orders_31_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_31/orders_31_2.dat.sorted
/flats/F1_31/orders_31_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_31/orders_31_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_31/orders_31_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_31/orders_31_3.dat >
/flats/F1_31/orders_31_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_31/orders_31_3.dat.sorted
/flats/F1_31/orders_31_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_31/orders_31_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_31/orders_31_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_31/orders_31_4.dat >
/flats/F1_31/orders_31_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_31/orders_31_4.dat.sorted
/flats/F1_31/orders_31_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_31/orders_31_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_31/orders_31_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_32/orders_32_1.dat >
/flats/F1_32/orders_32_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_32/orders_32_1.dat.sorted
/flats/F1_32/orders_32_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_32/orders_32_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_32/orders_32_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_32/orders_32_2.dat >
/flats/F1_32/orders_32_2.dat.sorted

```

```

if [ $? = 0 ]
then
  mv /flats/F1_32/orders_32_2.dat.sorted
/flats/F1_32/orders_32_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_32/orders_32_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_32/orders_32_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_32/orders_32_3.dat >
/flats/F1_32/orders_32_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_32/orders_32_3.dat.sorted
/flats/F1_32/orders_32_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_32/orders_32_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_32/orders_32_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_32/orders_32_4.dat >
/flats/F1_32/orders_32_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_32/orders_32_4.dat.sorted
/flats/F1_32/orders_32_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_32/orders_32_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_32/orders_32_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_33/orders_33_1.dat >
/flats/F1_33/orders_33_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_33/orders_33_1.dat.sorted
/flats/F1_33/orders_33_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_33/orders_33_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_33/orders_33_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_33/orders_33_2.dat >
/flats/F1_33/orders_33_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_33/orders_33_2.dat.sorted
/flats/F1_33/orders_33_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_33/orders_33_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_33/orders_33_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_33/orders_33_3.dat >
/flats/F1_33/orders_33_3.dat.sorted
if [ $? = 0 ]
then

```

```

  mv /flats/F1_33/orders_33_3.dat.sorted
/flats/F1_33/orders_33_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_33/orders_33_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_33/orders_33_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_33/orders_33_4.dat >
/flats/F1_33/orders_33_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_33/orders_33_4.dat.sorted
/flats/F1_33/orders_33_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_33/orders_33_4.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_33/orders_33_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_34/orders_34_1.dat >
/flats/F1_34/orders_34_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_34/orders_34_1.dat.sorted
/flats/F1_34/orders_34_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_34/orders_34_1.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_34/orders_34_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_34/orders_34_2.dat >
/flats/F1_34/orders_34_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_34/orders_34_2.dat.sorted
/flats/F1_34/orders_34_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_34/orders_34_2.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_34/orders_34_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_34/orders_34_3.dat >
/flats/F1_34/orders_34_3.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_34/orders_34_3.dat.sorted
/flats/F1_34/orders_34_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/F1_34/orders_34_3.dat FAILED."
  echo "Please ensure that the source file
(/flats/F1_34/orders_34_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_34/orders_34_4.dat >
/flats/F1_34/orders_34_4.dat.sorted
if [ $? = 0 ]
then
  mv /flats/F1_34/orders_34_4.dat.sorted
/flats/F1_34/orders_34_4.dat

```

```

else
echo "**** WARNING ****"
echo "Sort of /flats/F1_34/orders_34_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_34/orders_34_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_35/orders_35_1.dat >
/flats/F1_35/orders_35_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_35/orders_35_1.dat.sorted
/flats/F1_35/orders_35_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_35/orders_35_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_35/orders_35_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_35/orders_35_2.dat >
/flats/F1_35/orders_35_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_35/orders_35_2.dat.sorted
/flats/F1_35/orders_35_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_35/orders_35_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_35/orders_35_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_35/orders_35_3.dat >
/flats/F1_35/orders_35_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_35/orders_35_3.dat.sorted
/flats/F1_35/orders_35_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_35/orders_35_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_35/orders_35_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_35/orders_35_4.dat >
/flats/F1_35/orders_35_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_35/orders_35_4.dat.sorted
/flats/F1_35/orders_35_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_35/orders_35_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_35/orders_35_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_36/orders_36_1.dat >
/flats/F1_36/orders_36_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_36/orders_36_1.dat.sorted
/flats/F1_36/orders_36_1.dat

```

```

echo "**** WARNING ****"
echo "Sort of /flats/F1_36/orders_36_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_36/orders_36_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_36/orders_36_2.dat >
/flats/F1_36/orders_36_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_36/orders_36_2.dat.sorted
/flats/F1_36/orders_36_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_36/orders_36_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_36/orders_36_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_36/orders_36_3.dat >
/flats/F1_36/orders_36_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_36/orders_36_3.dat.sorted
/flats/F1_36/orders_36_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_36/orders_36_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_36/orders_36_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_36/orders_36_4.dat >
/flats/F1_36/orders_36_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_36/orders_36_4.dat.sorted
/flats/F1_36/orders_36_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_36/orders_36_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_36/orders_36_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_37/orders_37_1.dat >
/flats/F1_37/orders_37_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_37/orders_37_1.dat.sorted
/flats/F1_37/orders_37_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_37/orders_37_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_37/orders_37_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_37/orders_37_2.dat >
/flats/F1_37/orders_37_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_37/orders_37_2.dat.sorted
/flats/F1_37/orders_37_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_37/orders_37_2.dat FAILED."

```

```

    echo "Please ensure that the source file
(flats/F1_37/orders_37_2.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_37/orders_37_3.dat >
/flats/F1_37/orders_37_3.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_37/orders_37_3.dat.sorted
/flats/F1_37/orders_37_3.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_37/orders_37_3.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_37/orders_37_3.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_37/orders_37_4.dat >
/flats/F1_37/orders_37_4.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_37/orders_37_4.dat.sorted
/flats/F1_37/orders_37_4.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_37/orders_37_4.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_37/orders_37_4.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_38/orders_38_1.dat >
/flats/F1_38/orders_38_1.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_38/orders_38_1.dat.sorted
/flats/F1_38/orders_38_1.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_38/orders_38_1.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_38/orders_38_1.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_38/orders_38_2.dat >
/flats/F1_38/orders_38_2.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_38/orders_38_2.dat.sorted
/flats/F1_38/orders_38_2.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_38/orders_38_2.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_38/orders_38_2.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_38/orders_38_3.dat >
/flats/F1_38/orders_38_3.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_38/orders_38_3.dat.sorted
/flats/F1_38/orders_38_3.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_38/orders_38_3.dat FAILED."

```

```

    echo "Please ensure that the source file
(flats/F1_38/orders_38_3.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_38/orders_38_4.dat >
/flats/F1_38/orders_38_4.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_38/orders_38_4.dat.sorted
/flats/F1_38/orders_38_4.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_38/orders_38_4.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_38/orders_38_4.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_39/orders_39_1.dat >
/flats/F1_39/orders_39_1.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_39/orders_39_1.dat.sorted
/flats/F1_39/orders_39_1.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_39/orders_39_1.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_39/orders_39_1.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_39/orders_39_2.dat >
/flats/F1_39/orders_39_2.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_39/orders_39_2.dat.sorted
/flats/F1_39/orders_39_2.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_39/orders_39_2.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_39/orders_39_2.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_39/orders_39_3.dat >
/flats/F1_39/orders_39_3.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_39/orders_39_3.dat.sorted
/flats/F1_39/orders_39_3.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_39/orders_39_3.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_39/orders_39_3.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_39/orders_39_4.dat >
/flats/F1_39/orders_39_4.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_39/orders_39_4.dat.sorted
/flats/F1_39/orders_39_4.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_39/orders_39_4.dat FAILED."
    echo "Please ensure that the source file
(flats/F1_39/orders_39_4.dat)"

```

```

echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_40/orders_40_1.dat >
/flats/F1_40/orders_40_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_40/orders_40_1.dat.sorted
/flats/F1_40/orders_40_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_40/orders_40_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_40/orders_40_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_40/orders_40_2.dat >
/flats/F1_40/orders_40_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_40/orders_40_2.dat.sorted
/flats/F1_40/orders_40_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_40/orders_40_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_40/orders_40_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_40/orders_40_3.dat >
/flats/F1_40/orders_40_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_40/orders_40_3.dat.sorted
/flats/F1_40/orders_40_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_40/orders_40_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_40/orders_40_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n /flats/F1_40/orders_40_4.dat >
/flats/F1_40/orders_40_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_40/orders_40_4.dat.sorted
/flats/F1_40/orders_40_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_40/orders_40_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_40/orders_40_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

SORT NEW ORDER.sh

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_01/neworder_01_1.dat
> /flats/F1_01/neworder_01_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_01/neworder_01_1.dat.sorted
/flats/F1_01/neworder_01_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_01/neworder_01_1.dat FAILED."

```

```

echo "Please ensure that the source file
(flats/F1_01/neworder_01_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_01/neworder_01_2.dat
> /flats/F1_01/neworder_01_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_01/neworder_01_2.dat.sorted
/flats/F1_01/neworder_01_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_01/neworder_01_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_01/neworder_01_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_01/neworder_01_3.dat
> /flats/F1_01/neworder_01_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_01/neworder_01_3.dat.sorted
/flats/F1_01/neworder_01_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_01/neworder_01_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_01/neworder_01_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_01/neworder_01_4.dat
> /flats/F1_01/neworder_01_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_01/neworder_01_4.dat.sorted
/flats/F1_01/neworder_01_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_01/neworder_01_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_01/neworder_01_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_02/neworder_02_1.dat
> /flats/F1_02/neworder_02_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_02/neworder_02_1.dat.sorted
/flats/F1_02/neworder_02_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_02/neworder_02_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_02/neworder_02_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_02/neworder_02_2.dat
> /flats/F1_02/neworder_02_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_02/neworder_02_2.dat.sorted
/flats/F1_02/neworder_02_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_02/neworder_02_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_02/neworder_02_2.dat)"

```

```

    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_02/neworder_02_3.dat
> /flats/F1_02/neworder_02_3.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_02/neworder_02_3.dat.sorted
/flats/F1_02/neworder_02_3.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_02/neworder_02_3.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_02/neworder_02_3.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_02/neworder_02_4.dat
> /flats/F1_02/neworder_02_4.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_02/neworder_02_4.dat.sorted
/flats/F1_02/neworder_02_4.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_02/neworder_02_4.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_02/neworder_02_4.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_03/neworder_03_1.dat
> /flats/F1_03/neworder_03_1.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_03/neworder_03_1.dat.sorted
/flats/F1_03/neworder_03_1.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_03/neworder_03_1.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_03/neworder_03_1.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_03/neworder_03_2.dat
> /flats/F1_03/neworder_03_2.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_03/neworder_03_2.dat.sorted
/flats/F1_03/neworder_03_2.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_03/neworder_03_2.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_03/neworder_03_2.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_03/neworder_03_3.dat
> /flats/F1_03/neworder_03_3.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_03/neworder_03_3.dat.sorted
/flats/F1_03/neworder_03_3.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_03/neworder_03_3.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_03/neworder_03_3.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."

```

```

fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_03/neworder_03_4.dat
> /flats/F1_03/neworder_03_4.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_03/neworder_03_4.dat.sorted
/flats/F1_03/neworder_03_4.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_03/neworder_03_4.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_03/neworder_03_4.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_04/neworder_04_1.dat
> /flats/F1_04/neworder_04_1.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_04/neworder_04_1.dat.sorted
/flats/F1_04/neworder_04_1.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_04/neworder_04_1.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_04/neworder_04_1.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_04/neworder_04_2.dat
> /flats/F1_04/neworder_04_2.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_04/neworder_04_2.dat.sorted
/flats/F1_04/neworder_04_2.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_04/neworder_04_2.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_04/neworder_04_2.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_04/neworder_04_3.dat
> /flats/F1_04/neworder_04_3.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_04/neworder_04_3.dat.sorted
/flats/F1_04/neworder_04_3.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_04/neworder_04_3.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_04/neworder_04_3.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_04/neworder_04_4.dat
> /flats/F1_04/neworder_04_4.dat.sorted
if [ $? = 0 ]
then
    mv /flats/F1_04/neworder_04_4.dat.sorted
/flats/F1_04/neworder_04_4.dat
else
    echo "**** WARNING ****"
    echo "Sort of /flats/F1_04/neworder_04_4.dat FAILED."
    echo "Please ensure that the source file
(/flats/F1_04/neworder_04_4.dat)"
    echo "exists and that sufficient free space is available"
    echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_05/neworder_05_1.dat
> /flats/F1_05/neworder_05_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/neworder_05_1.dat.sorted
/flats/F1_05/neworder_05_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/neworder_05_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/neworder_05_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_05/neworder_05_2.dat
> /flats/F1_05/neworder_05_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/neworder_05_2.dat.sorted
/flats/F1_05/neworder_05_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/neworder_05_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/neworder_05_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_05/neworder_05_3.dat
> /flats/F1_05/neworder_05_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/neworder_05_3.dat.sorted
/flats/F1_05/neworder_05_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/neworder_05_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/neworder_05_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_05/neworder_05_4.dat
> /flats/F1_05/neworder_05_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_05/neworder_05_4.dat.sorted
/flats/F1_05/neworder_05_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_05/neworder_05_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_05/neworder_05_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_06/neworder_06_1.dat
> /flats/F1_06/neworder_06_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/neworder_06_1.dat.sorted
/flats/F1_06/neworder_06_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/neworder_06_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/neworder_06_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_06/neworder_06_2.dat
> /flats/F1_06/neworder_06_2.dat.sorted

```

```

if [ $? = 0 ]
then
mv /flats/F1_06/neworder_06_2.dat.sorted
/flats/F1_06/neworder_06_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/neworder_06_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/neworder_06_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_06/neworder_06_3.dat
> /flats/F1_06/neworder_06_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/neworder_06_3.dat.sorted
/flats/F1_06/neworder_06_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/neworder_06_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/neworder_06_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_06/neworder_06_4.dat
> /flats/F1_06/neworder_06_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_06/neworder_06_4.dat.sorted
/flats/F1_06/neworder_06_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_06/neworder_06_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_06/neworder_06_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_07/neworder_07_1.dat
> /flats/F1_07/neworder_07_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/neworder_07_1.dat.sorted
/flats/F1_07/neworder_07_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_07/neworder_07_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/neworder_07_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_07/neworder_07_2.dat
> /flats/F1_07/neworder_07_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/neworder_07_2.dat.sorted
/flats/F1_07/neworder_07_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_07/neworder_07_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/neworder_07_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_07/neworder_07_3.dat
> /flats/F1_07/neworder_07_3.dat.sorted
if [ $? = 0 ]
then

```



```

mv /flats/F1_07/neworder_07_3.dat.sorted
/flats/F1_07/neworder_07_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_07/neworder_07_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/neworder_07_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_07/neworder_07_4.dat
> /flats/F1_07/neworder_07_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_07/neworder_07_4.dat.sorted
/flats/F1_07/neworder_07_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_07/neworder_07_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_07/neworder_07_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_08/neworder_08_1.dat
> /flats/F1_08/neworder_08_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/neworder_08_1.dat.sorted
/flats/F1_08/neworder_08_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_08/neworder_08_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/neworder_08_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_08/neworder_08_2.dat
> /flats/F1_08/neworder_08_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/neworder_08_2.dat.sorted
/flats/F1_08/neworder_08_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_08/neworder_08_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/neworder_08_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_08/neworder_08_3.dat
> /flats/F1_08/neworder_08_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/neworder_08_3.dat.sorted
/flats/F1_08/neworder_08_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_08/neworder_08_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/neworder_08_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n /flats/F1_08/neworder_08_4.dat
> /flats/F1_08/neworder_08_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_08/neworder_08_4.dat.sorted
/flats/F1_08/neworder_08_4.dat

```

```

else
echo "**** WARNING ****"
echo "Sort of /flats/F1_08/neworder_08_4.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_08/neworder_08_4.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

C.2 Data Generation Code

dbgen/Makefile

```

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

include $(TPCC_ROOT)/Makefile.config

#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####

INCLUDE = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include

CFLAGS = $(INCLUDE) $(CFLAGS_OS) -DLINT_ARGS -
DSQLA_NOLINES \
-D$(DB2EDITION) -D$(DB2VERSION)
$(CFLAGS_DEBUG)

LDFLAGS = $(LDFLAGS_EXEC) $(LDFLAGS_LIB)

#
#####
#####
# File Collections
#
#####
#####

OBJ = tpcrnd$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE = $(TPCC_ROOT)/Src.Common/tpccclwh$(OBJEXT)

EXEC = gendata$(BINEXT)

#
#####
#####
# End-User Targets

```

```

#
#####
#####

all: $(EXEC)

clean:
    - $(ERASE) *$(OBJEXT) $(EXEC)

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

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

# We use $$$(OBJEXT) here so that the UNIX makefiles work with
both
# 'traditional' make and GNU make
$(EXEC):
    $(LD_EXEC) $(LDFLAGS) $(OBJS) $$$(OBJEXT)
$(LDFLAGS_OUT)$$@

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

# Link Dependencies
gendata$(BINEXT): $(OBJS) gendata$(OBJEXT)

# Build Dependencies
# Source
gendata$(OBJEXT):

# 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|%d|%ld|%d\n";
char fmtDist[]  = "%d|%d|%ld|%s|%s|%s|%s|%s|%s|%d|%d\n";
char fmtItem[]  = "%s|%d|%s|%d|%d\n";
char fmtStock[] =
"%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%d|%d\n";
char fmtCust[]  =
"%d|%s|%s|%s|%ld|%ld|%s|%s|%d|%s|%s|%s|%s|%s|%d|%d|
%d|%ld|%ld|%d\n";
char fmtHist[]  = "%d|%d|%d|%d|%d|%ld|%d|%d|%s\n";
char fmtOrdr[]  = "%d|%ld|%d|%d|%d|%d|%d|%d\n";
char fmtOLine[] = "%ld|%d|%d|%d|%d|%s|%d|%d|%d|%d\n";
char fmtNewOrd[] = "%d|%d|%d\n";
void InitFormatStrings(char delim);
void ScalingReport(void);

int outtype1 = 0;
int outtype2 = 0;
char *outname1 = NULL;
char *outname2 = NULL;

/*-----*/
/* main */
/*-----*/
int main (int argc, char *argv[])
{
    int option = -1;
    char *delim = NULL;

    /* ***** */
    /* Compute Warehouse Ranges */
    /* ***** */
    ware_start = 1;
    ware_end = WAREHOUSES;

    /* ***** */
    /* Process Command Line Arguments */
    /* ***** */

    /* Valid Command Line Options
    *-----

```

```

* Table Option:      -t <table>      (-t 3 for warehouse)
* Output Column Delimiter: -d <char>      (-d '|', -d '\', etc)
* Output to File:    -f[n] <file>      (-f customer.dat)
* Output to Pipe:   -p[n] <pipename>  (-p tpccpipe.000)
* Warehouse Range:  -r <start> <end>  (-r 1 100)
* Scaling Report:   -s
* Quiet Mode:       -q
*
* The -f[n] and/or -p[n] options are required.
* The -t, -d, -r, -s and -q options are optional.
*
* If -d is omitted, the vertical bar (pipe) symbol (|) will be used.
* If -r is omitted, the range [1..WAREHOUSES] will be used.
*
* Due to the TPC-C spec requiring that orders and orderline be
* generated at the same time, there is an extension to the -f and -
p
* options to specify one of the two output streams for each
argument.
*
* -f1 orders.dat -f2 orderline.dat will output to two files
* -f1 orders.dat -p2 tpccpipe.000 will output to a file and a pipe
*
* -f1/-p1 specifies the destination for the orders table
* -f2/-p2 specifies the destination for the orderline table
*
*/

/* Read Arguments */
for (i=1; i<argc; i++)
{
    if (strcmp(argv[i], "-t") == 0) {
        option = atoi(argv[i+1]);
        i++;
    } else if (strcmp(argv[i], "-r") == 0) {
        ware_start = atoi(argv[i+1]);
        ware_end = atoi(argv[i+2]);
        i += 2;
    } else if (strcmp(argv[i], "-d") == 0) {
        delim = argv[i+1];
        i++;
    } else if ((strcmp(argv[i], "-f") == 0) ||
                (strcmp(argv[i], "-f1") == 0)) {
        outtype1 = IOH_FILE;
        outname1 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-f2") == 0) {
        outtype2 = IOH_FILE;
        outname2 = argv[i+1];
        i++;
    } else if ((strcmp(argv[i], "-p") == 0) ||
                (strcmp(argv[i], "-p1") == 0)) {
        outtype1 = IOH_PIPE;
        outname1 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-p2") == 0) {
        outtype2 = IOH_PIPE;
        outname2 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-s") == 0) {
        ScalingReport();
        exit(0);
    } else if (strcmp(argv[i], "-q") == 0) {
        quiet_mode = 1;
    } else {
        fprintf(stderr, "gendata: Don't understand argument:
%s\n", argv[i]);
        exit(-1);
    }
}

/* ***** */

```

```

/* Validate Command Line Arguments */
/* ***** */

/* Validate Table Argument */
if (option < 3 || option > 11 || option == 10)
{
    fprintf(stderr, "gendata: Invalid table selected: %d\n", option);
    exit(-1);
}

/* Validate Delimiter Argument */
if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes
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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nSTOCK table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nSTOCK table FAILED at (S %d W %d) after
%8.2f seconds.\n\n", stock_num, ware_num, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate warehouse table */
/*-----*/
void gen_ware_tbl( void )
{
    sqlint32 ware_num = 0 ;
    char ware_name[11] ;
    char ware_street_1[21] ;
    char ware_street_2[21] ;
    char ware_city[21] ;
    char ware_state[3] ;
    char ware_zip[10] ;
    sqlint32 ware_tax ;

```

```

sqlint64 ware_YTD ;

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];

initialize_random(23,111);
timestamp1 = current_time();

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto ware_done; }

for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
{
    if (!quiet_mode && ((ware_num % 500) == 0))
        //@dxxxxxmt
        fprintf(stdout, "Warehouse #%d\n", ware_num);
        fflush(stdout);
}

create_random_a_string( ware_name, 6,10); /* create name
*/
create_random_a_string( ware_street_1, 10,20); /* create street
1 */
create_random_a_string( ware_street_2, 10,20); /* create street
2 */
create_random_a_string( ware_city, 10,20); /* create city */
create_random_a_string( ware_state, 2,2); /* create state */
create_random_n_string( ware_zip, 4,4); /* create zip */
strcat(ware_zip, "11111");

ware_tax = rand_integer(0, 2000);
ware_YTD = 30000000;

numBytes = sprintf(Buffer, fmtWare,
    ware_name,
    ware_street_1,
    ware_street_2,
    ware_city,
    ware_state,
    ware_zip,
    ware_tax,
    ware_YTD,
    ware_num);

rc = GenericWrite(&hnd, Buffer, numBytes);
if (rc != 0) { goto ware_done; }

} /* end for */

rc = GenericClose(&hnd);

ware_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after
%8.2f seconds.\n\n", ware_num, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate dist table */
/*-----*/
void gen_dist_tbl( void )

```

```

{
  sqlint32 ware_num = 0 ;
  sqlint32 dist_num = 0 ;
  char dist_name[11];
  char dist_street_1[21];
  char dist_street_2[21];
  char dist_city[21];
  char dist_state[3];
  char dist_zip[10];
  sqlint32 dist_tax;
  sqlint32 next_o_id;
  sqlint64 dist_YTD;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  next_o_id = CUSTOMERS_PER_DISTRICT + 1;
  initialize_random(44,73);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto dist_done; }

  for (ware_num = ware_start; ware_num <= ware_end;
  ware_num++)
  {
    if (!quiet_mode) {
      fprintf(stdout, "DISTRICT for Warehouse #%d\n", ware_num);
      fflush(stdout);
    }
    for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
      create_random_a_string( dist_name,    6,10) ; /* create name
*/
      create_random_a_string( dist_street_1, 10,20) ; /* create
street 1 */
      create_random_a_string( dist_street_2, 10,20) ; /* create
street 2 */
      create_random_a_string( dist_city,    10,20) ; /* create city */
      create_random_a_string( dist_state,    2,2) ; /* create state */
      create_random_n_string( dist_zip,      4,4) ; /* create zip */
      strcat(dist_zip, "11111");
      dist_tax = rand_integer(0, 2000);
      dist_YTD = 3000000;

      numBytes = sprintf(Buffer, fmtDist,
        next_o_id,
        dist_tax,
        dist_YTD,
        dist_name,
        dist_street_1,
        dist_street_2,
        dist_city,
        dist_state,
        dist_zip,
        dist_num,
        ware_num);

      rc = GenericWrite(&hnd, Buffer, numBytes);
      if (rc != 0) { goto dist_done; }

    } /* end for... */
  } /* end for... */

  rc = GenericClose(&hnd);

dist_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;

```

```

if (rc == 0) {
  fprintf(stdout, "\nDISTRICT table generated in %8.2f
seconds.\n\n", elapse);
  fflush(stdout);
} else {
  fprintf(stderr, "\nDISTRICT table FAILED at (W %d D %d) after
%8.2f seconds.\n\n", ware_num, dist_num, elapse);
  fflush(stderr);
}
}

/*-----*/
/* generate customer table
/*-----*/
void gen_cust_tbl( void )
{
  sqlint32 ware_num = 0 ;
  sqlint32 dist_num = 0 ;
  sqlint32 cust_num = 0 ;
  char cust_last[17];
  char cust_middle[3];
  char cust_first[17];
  char cust_street_1[21];
  char cust_street_2[21];
  char cust_city[21];
  char cust_state[3];
  char cust_zip[10];
  char cust_phone[17];
  char cust_credit[3];
  char cust_data[501];
  sqlint32 cust_discount;
  sqlint64 currtmstp;
  sqlint64 cust_balance;
  sqlint64 cust_YTD_payment;
  sqlint64 cust_credit_lim;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];
  int len, pos;

  initialize_random(10,64);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto cust_done; }

  strcpy(cust_middle, "OE");
  currtmstp = time(NULL);

  for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT;
cust_num++)
  {
    if (!quiet_mode) {
      fprintf(stdout, "CUSTOMER #%d:\n", cust_num);
      fflush(stdout);
    }

    for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
    {
      for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
      {
        if (cust_num <= 1000) /* create last name */
          create_random_last_name( cust_last, cust_num);
        else /* create last name */
          create_random_last_name( cust_last, 0);
        create_random_a_string( cust_first, 8,16) ; /* create first
name */
        create_random_a_string( cust_street_1, 10,20) ; /* create
street 1 */

```

```

        create_random_a_string( cust_street_2, 10,20) ; /* create
street 2 */
        create_random_a_string( cust_city,    10,20) ; /* create city
*/
        create_random_a_string( cust_state,   2,2) ; /* create state
*/
        create_random_n_string( cust_zip,    4,4) ; /* create zip */
        strcat(cust_zip, "11111");

        /* create phone number */
        create_random_n_string( cust_phone, 16,16) ;
        if ( rand_integer( 1, 100 ) <= 10 )
            strcpy( cust_credit, "BC" ) ;
        else
            strcpy( cust_credit, "GC" ) ;

        /* create discount rate */
        cust_discount = rand_integer(0, 5000);

        /* create customer data */
        create_random_a_string(cust_data, 300, 500);

        /* pad customer data (only for non-rctload) */
        if (using_rctload == 0) {
            for (pos=strlen(cust_data); pos<500; pos++)
                cust_data[pos] = ' ';
            cust_data[500] = '\0';
        }

        cust_credit_lim = 5000000;
        cust_balance = -1000;
        cust_YTD_payment = 1000;

        numBytes = sprintf(Buffer, fmtCust,
            cust_num,
            cust_state,
            cust_zip,
            cust_phone,
            currtmstmp,
            cust_credit_lim,
            cust_middle,
            cust_credit,
            cust_discount,
            cust_data,
            cust_last,
            cust_first,
            cust_street_1,
            cust_street_2,
            cust_city,
            dist_num,
            ware_num,
            0,
            cust_balance,
            cust_YTD_payment,
            1);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto cust_done; }

        } /* end for district... */
    } /* end for warehouse... */
} /* end for customer... */

rc = GenericClose(&hnd);

cust_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nCUSTOMER table generated in %8.2f
seconds.\n\n", elapse);

```

```

        fflush(stdout);
    } else {
        fprintf(stderr, "\nCUSTOMER table FAILED at (W %d D %d C
%d) after %8.2f seconds.\n\n", ware_num, dist_num, cust_num,
elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate hist table */
/*-----*/

void gen_hist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char hist_data[25] ;
    sqlint64 currtmstmp;

    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; }

    currtmstmp = 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,
                    currtmstmp,
                    1000,
                    hist_data);

                rc = GenericWrite(&hnd, Buffer, numBytes);
                if (rc != 0) { goto hist_done; }

            } /* end for customer... */
        } /* end for district... */
    } /* end for warehouse... */

    rc = GenericClose(&hnd);

hist_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {

```

```

    fprintf(stdout, "\nHISTORY table generated in %8.2f
seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nHISTORY table FAILED at (W %d D %d C %d)
after %8.2f seconds.\n\n", ware_num, dist_num, cust_num, elapse);
    fflush(stderr);
}
}

/*-----*/
/* generate nu_ord table */
/*-----*/
void gen_nu_ord_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 nu_ord_id = 0 ;
    int nu_ord_hi ;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    /* compute maximum and minimum
order numbers for this
district */
    nu_ord_hi = CUSTOMERS_PER_DISTRICT -
NU_ORDERS_PER_DISTRICT + 1;
    if (nu_ord_hi < 0) {
        nu_ord_hi = CUSTOMERS_PER_DISTRICT -
(CUSTOMERS_PER_DISTRICT / 3) + 1;
        fprintf(stderr, "\n**** WARNING ****
NU_ORDERS_PER_DISTRICT is >
CUSTOMERS_PER_DISTRICT\n");
        fprintf(stderr, "          Check the values in file lval.h\n");
        fprintf(stderr, "          Loading New-Order with 1/3 of
CUSTOMERS_PER_DISTRICT\n");
    }
    initialize_random(99,37);
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto neword_done; }

    for (nu_ord_id = nu_ord_hi;
nu_ord_id <= CUSTOMERS_PER_DISTRICT;
nu_ord_id++)
    {
        if (!quiet_mode) {
            fprintf(stdout, "NEW_ORDER for Customer #%-d:\n",
nu_ord_id);
            fflush(stdout);
        }
        for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
        {
            for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
            {
                numBytes = sprintf(Buffer, fmtNewOrd,
nu_ord_id,
dist_num,
ware_num);

                rc = GenericWrite(&hnd, Buffer, numBytes);
                if (rc != 0) { goto neword_done; }

            } /* end for... */
        } /* end for... */
    } /* end for... */
}

```

```

rc = GenericClose(&hnd);

neword_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nNEW_ORDER table FAILED at (W %d D %d O
%d) after %8.2f seconds.\n\n", ware_num, dist_num, nu_ord_id,
elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate order and order_line tables */
/*-----*/
void gen_ordr_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    sqlint32 ord_num = 0 ;
    sqlint32 ordr_carrier_id;
    sqlint32 ordr_ol_cnt;
    sqlint32 oline_ol_num;
    sqlint32 oline_item_num;

    sqlint32 oline_amount;
    char oline_dist_info[25];
    sqlint64 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();
  elapse = timestamp2 - timestamp1;
  if (rc1 == 0 && rc2 == 0) {
    fprintf(stdout, "\nORDERS & ORDER_LINE table(s) generated in
%8.2f seconds.\n\n", elapse);
    fflush(stdout);
  } else {
    fprintf(stderr, "\nORDERS & ORDER_LINE table(s) FAILED at
(W %d D %d O %d OL %d) after %8.2f seconds.\n\n", ware_num,
dist_num, ord_num, oline_ol_num, elapse);
    fflush(stderr);
  }
}

// This routine will initialize the printf format strings and replace the
// delimiter with the one provided. The pipe symbol is the default.
void InitFormatStrings(char delim)

```

```

{
  char *p;

  // Check if Using Default Delimiter
  if (delim == '|') return;

  // Replace Delimiters
  while (p = strchr(fmtWare, '|')) { *p = delim; }
  while (p = strchr(fmtDist, '|')) { *p = delim; }
  while (p = strchr(fmtItem, '|')) { *p = delim; }
  while (p = strchr(fmtStock, '|')) { *p = delim; }
  while (p = strchr(fmtCust, '|')) { *p = delim; }
  while (p = strchr(fmtHist, '|')) { *p = delim; }
  while (p = strchr(fmtOrdr, '|')) { *p = delim; }
  while (p = strchr(fmtOLine, '|')) { *p = delim; }
  while (p = strchr(fmtNewOrd, '|')) { *p = delim; }
}

void ScalingReport(void)
{
  /* Print Scaling Values */
  fprintf(stdout, "Scaling Values in Use\n");
  fprintf(stdout, "-----\n");
  fprintf(stdout, "Warehouses:          %d\n", WAREHOUSES);
  fprintf(stdout, "Districts/Warehouse: %d\n",
DISTRICTS_PER_WAREHOUSE);
  fprintf(stdout, "Customers/District:  %d\n",
CUSTOMERS_PER_DISTRICT);
  fprintf(stdout, "Items:                %d\n", ITEMS);
  fprintf(stdout, "Stock/Warehouse:     %d\n",
STOCK_PER_WAREHOUSE);
  fprintf(stdout, "Min Order Lines/Order: %d\n",
MIN_OL_PER_ORDER);
  fprintf(stdout, "Max Order Lines/Order: %d\n",
MAX_OL_PER_ORDER);
  fprintf(stdout, "New Orders/District: %d\n",
NU_ORDERS_PER_DISTRICT);
  fprintf(stdout, "-----\n");
}

dbgen/tpccrnd.c

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

/*
 * tpccrnd.c - Random generation functions for TPC-C
 */

#include "db2tpcc.h"
#include "lval.h"

static char tbl_cust[CUSTOMERS_PER_DISTRICT];

static char alnum[] =

"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
RSTUVWXYZ";

```

```

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);
}

/*
*****
*  NUrund_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 NUrund_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 = NUrand_val( A_C_LAST, 0, 999,
C_C_LAST_LOAD );
else
    random_num = cust_num - 1;

strcpy(out_buffer, last_name_parts[random_num / 100]);
random_num %= 100;
strcat(out_buffer, last_name_parts[random_num / 10]);
random_num %= 10;
strcat(out_buffer, last_name_parts[random_num]);

return(strlen(out_buffer));
}

```

include/lval.h

```

#ifndef __LVAL_H

#define __LVAL_H
#define WAREHOUSES 64000
#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

```
/* For Oracle in the tpcload program C_LAST=1. */
/* C-Delta be the difference between C-LOAD and C-Run. */
/* C-Delta must be a value between 65..119 including the */
/* values of 65 and 119 and excluding the value of 96 and 112 */

#define MASTER_NUM1 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

#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"
#endif

/*---- SUT -----*/
SUT="toraus"
/*-----*/
LASTC=88
MEASUREMENT="1"
WAREHOUSES=64000
/*----- SLAVES -----*/

#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c, driver1d, driver2a, driver2b,
driver2c, driver2d, driver3a, driver3b, driver3c, driver3d, driver4a,
driver4b, driver4c, driver4d
/*
SLAVES driver1a, driver1b, driver1c, driver1d
*/
#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
#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"
#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

#endif
/*-----*/
TELNET telnet 23
SOCKET socket 199703
/* --- Sockets -----*/
#if MASTER_NUM1
SOCKET_NETWORK socket1 80 driver1a
SOCKET_NETWORK socket2 80 driver1b
SOCKET_NETWORK socket3 80 driver1c
SOCKET_NETWORK socket4 80 driver1d
SOCKET_NETWORK socket5 80 driver1a
SOCKET_NETWORK socket6 80 driver1b
SOCKET_NETWORK socket7 80 driver1c
SOCKET_NETWORK socket8 80 driver1d
SOCKET_NETWORK socket9 80 driver1a
SOCKET_NETWORK socket10 80 driver1b
SOCKET_NETWORK socket11 80 driver1c
SOCKET_NETWORK socket12 80 driver1d
SOCKET_NETWORK socket13 80 driver1a
SOCKET_NETWORK socket14 80 driver1b
SOCKET_NETWORK socket15 80 driver1c
SOCKET_NETWORK socket16 80 driver1d
SOCKET_NETWORK socket17 80 driver1a
SOCKET_NETWORK socket18 80 driver1b
SOCKET_NETWORK socket19 80 driver1c
SOCKET_NETWORK socket20 80 driver1d
SOCKET_NETWORK socket21 80 driver1a
SOCKET_NETWORK socket22 80 driver1b
SOCKET_NETWORK socket23 80 driver1c
SOCKET_NETWORK socket24 80 driver1d
SOCKET_NETWORK socket25 80 driver1a
SOCKET_NETWORK socket26 80 driver1b
SOCKET_NETWORK socket27 80 driver1c
SOCKET_NETWORK socket28 80 driver1d
SOCKET_NETWORK socket29 80 driver1a
SOCKET_NETWORK socket30 80 driver1b
SOCKET_NETWORK socket31 80 driver1c
SOCKET_NETWORK socket32 80 driver1d
SOCKET_NETWORK socket33 80 driver2a
SOCKET_NETWORK socket34 80 driver2b
SOCKET_NETWORK socket35 80 driver2c
SOCKET_NETWORK socket36 80 driver2d
SOCKET_NETWORK socket37 80 driver2a
SOCKET_NETWORK socket38 80 driver2b
SOCKET_NETWORK socket39 80 driver2c
SOCKET_NETWORK socket40 80 driver2d
SOCKET_NETWORK socket41 80 driver2a
SOCKET_NETWORK socket42 80 driver2b
SOCKET_NETWORK socket43 80 driver2c
SOCKET_NETWORK socket44 80 driver2d
SOCKET_NETWORK socket45 80 driver2a
SOCKET_NETWORK socket46 80 driver2b
SOCKET_NETWORK socket47 80 driver2c
SOCKET_NETWORK socket48 80 driver2d
SOCKET_NETWORK socket49 80 driver2a
SOCKET_NETWORK socket50 80 driver2b
SOCKET_NETWORK socket51 80 driver2c
SOCKET_NETWORK socket52 80 driver2d
SOCKET_NETWORK socket53 80 driver2a
SOCKET_NETWORK socket54 80 driver2b
SOCKET_NETWORK socket55 80 driver2c
SOCKET_NETWORK socket56 80 driver2d
SOCKET_NETWORK socket57 80 driver2a
SOCKET_NETWORK socket58 80 driver2b
SOCKET_NETWORK socket59 80 driver2c
SOCKET_NETWORK socket60 80 driver2d
SOCKET_NETWORK socket61 80 driver2a
SOCKET_NETWORK socket62 80 driver2b
SOCKET_NETWORK socket63 80 driver2c
SOCKET_NETWORK socket64 80 driver2d

```



```

SOCKET_NETWORK socket922 80 driver29b
SOCKET_NETWORK socket923 80 driver29c
SOCKET_NETWORK socket924 80 driver29d
SOCKET_NETWORK socket925 80 driver29a
SOCKET_NETWORK socket926 80 driver29b
SOCKET_NETWORK socket927 80 driver29c
SOCKET_NETWORK socket928 80 driver29d
SOCKET_NETWORK socket929 80 driver30a
SOCKET_NETWORK socket930 80 driver30b
SOCKET_NETWORK socket931 80 driver30c
SOCKET_NETWORK socket932 80 driver30d
SOCKET_NETWORK socket933 80 driver30a
SOCKET_NETWORK socket934 80 driver30b
SOCKET_NETWORK socket935 80 driver30c
SOCKET_NETWORK socket936 80 driver30d
SOCKET_NETWORK socket937 80 driver30a
SOCKET_NETWORK socket938 80 driver30b
SOCKET_NETWORK socket939 80 driver30c
SOCKET_NETWORK socket940 80 driver30d
SOCKET_NETWORK socket941 80 driver30a
SOCKET_NETWORK socket942 80 driver30b
SOCKET_NETWORK socket943 80 driver30c
SOCKET_NETWORK socket944 80 driver30d
SOCKET_NETWORK socket945 80 driver30a
SOCKET_NETWORK socket946 80 driver30b
SOCKET_NETWORK socket947 80 driver30c
SOCKET_NETWORK socket948 80 driver30d
SOCKET_NETWORK socket949 80 driver30a
SOCKET_NETWORK socket950 80 driver30b
SOCKET_NETWORK socket951 80 driver30c
SOCKET_NETWORK socket952 80 driver30d
SOCKET_NETWORK socket953 80 driver30a
SOCKET_NETWORK socket954 80 driver30b
SOCKET_NETWORK socket955 80 driver30c
SOCKET_NETWORK socket956 80 driver30d
SOCKET_NETWORK socket957 80 driver30a
SOCKET_NETWORK socket958 80 driver30b
SOCKET_NETWORK socket959 80 driver30c
SOCKET_NETWORK socket960 80 driver30d
SOCKET_NETWORK socket961 80 driver31a
SOCKET_NETWORK socket962 80 driver31b
SOCKET_NETWORK socket963 80 driver31c
SOCKET_NETWORK socket964 80 driver31d
SOCKET_NETWORK socket965 80 driver31a
SOCKET_NETWORK socket966 80 driver31b
SOCKET_NETWORK socket967 80 driver31c
SOCKET_NETWORK socket968 80 driver31d
SOCKET_NETWORK socket969 80 driver31a
SOCKET_NETWORK socket970 80 driver31b
SOCKET_NETWORK socket971 80 driver31c
SOCKET_NETWORK socket972 80 driver31d
SOCKET_NETWORK socket973 80 driver31a
SOCKET_NETWORK socket974 80 driver31b
SOCKET_NETWORK socket975 80 driver31c
SOCKET_NETWORK socket976 80 driver31d
SOCKET_NETWORK socket977 80 driver31a
SOCKET_NETWORK socket978 80 driver31b
SOCKET_NETWORK socket979 80 driver31c
SOCKET_NETWORK socket980 80 driver31d
SOCKET_NETWORK socket981 80 driver31a
SOCKET_NETWORK socket982 80 driver31b
SOCKET_NETWORK socket983 80 driver31c
SOCKET_NETWORK socket984 80 driver31d
SOCKET_NETWORK socket985 80 driver31a
SOCKET_NETWORK socket986 80 driver31b
SOCKET_NETWORK socket987 80 driver31c
SOCKET_NETWORK socket988 80 driver31d
SOCKET_NETWORK socket989 80 driver31a
SOCKET_NETWORK socket990 80 driver31b
SOCKET_NETWORK socket991 80 driver31c
SOCKET_NETWORK socket992 80 driver31d
SOCKET_NETWORK socket993 80 driver32a

```

```

SOCKET_NETWORK socket994 80 driver32b
SOCKET_NETWORK socket995 80 driver32c
SOCKET_NETWORK socket996 80 driver32d
SOCKET_NETWORK socket997 80 driver32a
SOCKET_NETWORK socket998 80 driver32b
SOCKET_NETWORK socket999 80 driver32c
SOCKET_NETWORK socket1000 80 driver32d
SOCKET_NETWORK socket1001 80 driver32a
SOCKET_NETWORK socket1002 80 driver32b
SOCKET_NETWORK socket1003 80 driver32c
SOCKET_NETWORK socket1004 80 driver32d
SOCKET_NETWORK socket1005 80 driver32a
SOCKET_NETWORK socket1006 80 driver32b
SOCKET_NETWORK socket1007 80 driver32c
SOCKET_NETWORK socket1008 80 driver32d
SOCKET_NETWORK socket1009 80 driver32a
SOCKET_NETWORK socket1010 80 driver32b
SOCKET_NETWORK socket1011 80 driver32c
SOCKET_NETWORK socket1012 80 driver32d
SOCKET_NETWORK socket1013 80 driver32a
SOCKET_NETWORK socket1014 80 driver32b
SOCKET_NETWORK socket1015 80 driver32c
SOCKET_NETWORK socket1016 80 driver32d
SOCKET_NETWORK socket1017 80 driver32a
SOCKET_NETWORK socket1018 80 driver32b
SOCKET_NETWORK socket1019 80 driver32c
SOCKET_NETWORK socket1020 80 driver32d
SOCKET_NETWORK socket1021 80 driver32a
SOCKET_NETWORK socket1022 80 driver32b
SOCKET_NETWORK socket1023 80 driver32c
SOCKET_NETWORK socket1024 80 driver32d
#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=45:00
RUNTIME=150:00
RAMPDOWN_WAIT=2:00
RAMPDOWN=20:00
/*RAMPUP_SEC 2700 */
/*RUNTIME_SEC 9000 */
/*WAREHOUSES 64000 */
/*CLIENTS 960 */
#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 rtparams.null */
NEWORDER = "12.25, 0.42, 0.38"
PAYMENT = "12.25, 0.19, 0.23, 43.2, 41.1, 45.3 "
ORDSTAT = "10.50, 0.39, 0.21, 4.1, 3.9, 4.3 "

```

```

DELIVERY = "05.5, 0.19, 0.15, 4.1, 3.9, 4.3 "
STOCKLEV = "05.5, 0.25, 0.18, 4.1, 3.9, 4.3 "
#elif 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 client1b socket1 620 0-62
START_RANGE client1b socket2 630 62-125
START_RANGE client1b socket3 620 125-187
START_RANGE client1b socket4 630 187-250
START_RANGE client1b socket5 620 250-312
START_RANGE client1b socket6 630 312-375
START_RANGE client1b socket7 620 375-437
START_RANGE client1b socket8 630 437-500
START_RANGE client1b socket9 620 500-562
START_RANGE client1b socket10 630 562-625
START_RANGE client1b socket11 620 625-687
START_RANGE client1b socket12 630 687-750
START_RANGE client1b socket13 620 750-812
START_RANGE client1b socket14 630 812-875
START_RANGE client1b socket15 620 875-937
START_RANGE client1b socket16 630 937-1000
*/

START_RANGE client1a socket1 620 0-62
START_RANGE client1a socket2 630 62-125
START_RANGE client1a socket3 620 125-187
START_RANGE client1a socket4 630 187-250
START_RANGE client1a socket5 620 250-312
START_RANGE client1a socket6 630 312-375
START_RANGE client1a socket7 620 375-437
START_RANGE client1a socket8 630 437-500
START_RANGE client1a socket9 620 500-562
START_RANGE client1a socket10 630 562-625
START_RANGE client1a socket11 620 625-687
START_RANGE client1a socket12 630 687-750
START_RANGE client1a socket13 620 750-812
START_RANGE client1a socket14 630 812-875
START_RANGE client1a socket15 620 875-937
START_RANGE client1a socket16 630 937-1000

START_RANGE client1a socket17 620 1000-1062
START_RANGE client1a socket18 630 1062-1125
START_RANGE client1a socket19 620 1125-1187
START_RANGE client1a socket20 630 1187-1250
START_RANGE client1a socket21 620 1250-1312
START_RANGE client1a socket22 630 1312-1375
START_RANGE client1a socket23 620 1375-1437
START_RANGE client1a socket24 630 1437-1500
START_RANGE client1a socket25 620 1500-1562
START_RANGE client1a socket26 630 1562-1625
START_RANGE client1a socket27 620 1625-1687
START_RANGE client1a socket28 630 1687-1750
START_RANGE client1a socket29 620 1750-1812
START_RANGE client1a socket30 630 1812-1875
START_RANGE client1a socket31 620 1875-1937
START_RANGE client1a socket32 630 1937-2000

START_RANGE client2a socket33 620 2000-2062
START_RANGE client2a socket34 630 2062-2125
START_RANGE client2a socket35 620 2125-2187
START_RANGE client2a socket36 630 2187-2250
START_RANGE client2a socket37 620 2250-2312
START_RANGE client2a socket38 630 2312-2375
START_RANGE client2a socket39 620 2375-2437
START_RANGE client2a socket40 630 2437-2500

```

```

START_RANGE client2a socket41 620 2500-2562
START_RANGE client2a socket42 630 2562-2625
START_RANGE client2a socket43 620 2625-2687
START_RANGE client2a socket44 630 2687-2750
START_RANGE client2a socket45 620 2750-2812
START_RANGE client2a socket46 630 2812-2875
START_RANGE client2a socket47 620 2875-2937
START_RANGE client2a socket48 630 2937-3000

START_RANGE client2a socket49 620 3000-3062
START_RANGE client2a socket50 630 3062-3125
START_RANGE client2a socket51 620 3125-3187
START_RANGE client2a socket52 630 3187-3250
START_RANGE client2a socket53 620 3250-3312
START_RANGE client2a socket54 630 3312-3375
START_RANGE client2a socket55 620 3375-3437
START_RANGE client2a socket56 630 3437-3500
START_RANGE client2a socket57 620 3500-3562
START_RANGE client2a socket58 630 3562-3625
START_RANGE client2a socket59 620 3625-3687
START_RANGE client2a socket60 630 3687-3750
START_RANGE client2a socket61 620 3750-3812
START_RANGE client2a socket62 630 3812-3875
START_RANGE client2a socket63 620 3875-3937
START_RANGE client2a socket64 630 3937-4000

START_RANGE client3a socket65 620 4000-4062
START_RANGE client3a socket66 630 4062-4125
START_RANGE client3a socket67 620 4125-4187
START_RANGE client3a socket68 630 4187-4250
START_RANGE client3a socket69 620 4250-4312
START_RANGE client3a socket70 630 4312-4375
START_RANGE client3a socket71 620 4375-4437
START_RANGE client3a socket72 630 4437-4500
START_RANGE client3a socket73 620 4500-4562
START_RANGE client3a socket74 630 4562-4625
START_RANGE client3a socket75 620 4625-4687
START_RANGE client3a socket76 630 4687-4750
START_RANGE client3a socket77 620 4750-4812
START_RANGE client3a socket78 630 4812-4875
START_RANGE client3a socket79 620 4875-4937
START_RANGE client3a socket80 630 4937-5000

START_RANGE client3a socket81 620 5000-5062
START_RANGE client3a socket82 630 5062-5125
START_RANGE client3a socket83 620 5125-5187
START_RANGE client3a socket84 630 5187-5250
START_RANGE client3a socket85 620 5250-5312
START_RANGE client3a socket86 630 5312-5375
START_RANGE client3a socket87 620 5375-5437
START_RANGE client3a socket88 630 5437-5500
START_RANGE client3a socket89 620 5500-5562
START_RANGE client3a socket90 630 5562-5625
START_RANGE client3a socket91 620 5625-5687
START_RANGE client3a socket92 630 5687-5750
START_RANGE client3a socket93 620 5750-5812
START_RANGE client3a socket94 630 5812-5875
START_RANGE client3a socket95 620 5875-5937
START_RANGE client3a socket96 630 5937-6000

START_RANGE client4a socket97 620 6000-6062
START_RANGE client4a socket98 630 6062-6125
START_RANGE client4a socket99 620 6125-6187
START_RANGE client4a socket100 630 6187-6250
START_RANGE client4a socket101 620 6250-6312
START_RANGE client4a socket102 630 6312-6375
START_RANGE client4a socket103 620 6375-6437
START_RANGE client4a socket104 630 6437-6500
START_RANGE client4a socket105 620 6500-6562
START_RANGE client4a socket106 630 6562-6625
START_RANGE client4a socket107 620 6625-6687
START_RANGE client4a socket108 630 6687-6750

```


START_RANGE client4a socket109 620 6750-6812
START_RANGE client4a socket110 630 6812-6875
START_RANGE client4a socket111 620 6875-6937
START_RANGE client4a socket112 630 6937-7000

START_RANGE client4a socket113 620 7000-7062
START_RANGE client4a socket114 630 7062-7125
START_RANGE client4a socket115 620 7125-7187
START_RANGE client4a socket116 630 7187-7250
START_RANGE client4a socket117 620 7250-7312
START_RANGE client4a socket118 630 7312-7375
START_RANGE client4a socket119 620 7375-7437
START_RANGE client4a socket120 630 7437-7500
START_RANGE client4a socket121 620 7500-7562
START_RANGE client4a socket122 630 7562-7625
START_RANGE client4a socket123 620 7625-7687
START_RANGE client4a socket124 630 7687-7750
START_RANGE client4a socket125 620 7750-7812
START_RANGE client4a socket126 630 7812-7875
START_RANGE client4a socket127 620 7875-7937
START_RANGE client4a socket128 630 7937-8000

#elif MASTER_NUM2

START_RANGE client5a socket129 620 8000-8062
START_RANGE client5a socket130 630 8062-8125
START_RANGE client5a socket131 620 8125-8187
START_RANGE client5a socket132 630 8187-8250
START_RANGE client5a socket133 620 8250-8312
START_RANGE client5a socket134 630 8312-8375
START_RANGE client5a socket135 620 8375-8437
START_RANGE client5a socket136 630 8437-8500
START_RANGE client5a socket137 620 8500-8562
START_RANGE client5a socket138 630 8562-8625
START_RANGE client5a socket139 620 8625-8687
START_RANGE client5a socket140 630 8687-8750
START_RANGE client5a socket141 620 8750-8812
START_RANGE client5a socket142 630 8812-8875
START_RANGE client5a socket143 620 8875-8937
START_RANGE client5a socket144 630 8937-9000

START_RANGE client5a socket145 620 9000-9062
START_RANGE client5a socket146 630 9062-9125
START_RANGE client5a socket147 620 9125-9187
START_RANGE client5a socket148 630 9187-9250
START_RANGE client5a socket149 620 9250-9312
START_RANGE client5a socket150 630 9312-9375
START_RANGE client5a socket151 620 9375-9437
START_RANGE client5a socket152 630 9437-9500
START_RANGE client5a socket153 620 9500-9562
START_RANGE client5a socket154 630 9562-9625
START_RANGE client5a socket155 620 9625-9687
START_RANGE client5a socket156 630 9687-9750
START_RANGE client5a socket157 620 9750-9812
START_RANGE client5a socket158 630 9812-9875
START_RANGE client5a socket159 620 9875-9937
START_RANGE client5a socket160 630 9937-10000

START_RANGE client6a socket161 620 10000-10062
START_RANGE client6a socket162 630 10062-10125
START_RANGE client6a socket163 620 10125-10187
START_RANGE client6a socket164 630 10187-10250
START_RANGE client6a socket165 620 10250-10312
START_RANGE client6a socket166 630 10312-10375
START_RANGE client6a socket167 620 10375-10437
START_RANGE client6a socket168 630 10437-10500
START_RANGE client6a socket169 620 10500-10562
START_RANGE client6a socket170 630 10562-10625
START_RANGE client6a socket171 620 10625-10687
START_RANGE client6a socket172 630 10687-10750
START_RANGE client6a socket173 620 10750-10812
START_RANGE client6a socket174 630 10812-10875
START_RANGE client6a socket175 620 10875-10937

START_RANGE client6a socket176 630 10937-11000

START_RANGE client6a socket177 620 11000-11062
START_RANGE client6a socket178 630 11062-11125
START_RANGE client6a socket179 620 11125-11187
START_RANGE client6a socket180 630 11187-11250
START_RANGE client6a socket181 620 11250-11312
START_RANGE client6a socket182 630 11312-11375
START_RANGE client6a socket183 620 11375-11437
START_RANGE client6a socket184 630 11437-11500
START_RANGE client6a socket185 620 11500-11562
START_RANGE client6a socket186 630 11562-11625
START_RANGE client6a socket187 620 11625-11687
START_RANGE client6a socket188 630 11687-11750
START_RANGE client6a socket189 620 11750-11812
START_RANGE client6a socket190 630 11812-11875
START_RANGE client6a socket191 620 11875-11937
START_RANGE client6a socket192 630 11937-12000

START_RANGE client7a socket193 620 12000-12062
START_RANGE client7a socket194 630 12062-12125
START_RANGE client7a socket195 620 12125-12187
START_RANGE client7a socket196 630 12187-12250
START_RANGE client7a socket197 620 12250-12312
START_RANGE client7a socket198 630 12312-12375
START_RANGE client7a socket199 620 12375-12437
START_RANGE client7a socket200 630 12437-12500
START_RANGE client7a socket201 620 12500-12562
START_RANGE client7a socket202 630 12562-12625
START_RANGE client7a socket203 620 12625-12687
START_RANGE client7a socket204 630 12687-12750
START_RANGE client7a socket205 620 12750-12812
START_RANGE client7a socket206 630 12812-12875
START_RANGE client7a socket207 620 12875-12937
START_RANGE client7a socket208 630 12937-13000

START_RANGE client7a socket209 620 13000-13062
START_RANGE client7a socket210 630 13062-13125
START_RANGE client7a socket211 620 13125-13187
START_RANGE client7a socket212 630 13187-13250
START_RANGE client7a socket213 620 13250-13312
START_RANGE client7a socket214 630 13312-13375
START_RANGE client7a socket215 620 13375-13437
START_RANGE client7a socket216 630 13437-13500
START_RANGE client7a socket217 620 13500-13562
START_RANGE client7a socket218 630 13562-13625
START_RANGE client7a socket219 620 13625-13687
START_RANGE client7a socket220 630 13687-13750
START_RANGE client7a socket221 620 13750-13812
START_RANGE client7a socket222 630 13812-13875
START_RANGE client7a socket223 620 13875-13937
START_RANGE client7a socket224 630 13937-14000

START_RANGE client8a socket225 620 14000-14062
START_RANGE client8a socket226 630 14062-14125
START_RANGE client8a socket227 620 14125-14187
START_RANGE client8a socket228 630 14187-14250
START_RANGE client8a socket229 620 14250-14312
START_RANGE client8a socket230 630 14312-14375
START_RANGE client8a socket231 620 14375-14437
START_RANGE client8a socket232 630 14437-14500
START_RANGE client8a socket233 620 14500-14562
START_RANGE client8a socket234 630 14562-14625
START_RANGE client8a socket235 620 14625-14687
START_RANGE client8a socket236 630 14687-14750
START_RANGE client8a socket237 620 14750-14812
START_RANGE client8a socket238 630 14812-14875
START_RANGE client8a socket239 620 14875-14937
START_RANGE client8a socket240 630 14937-15000

START_RANGE client8a socket241 620 15000-15062
START_RANGE client8a socket242 630 15062-15125

START_RANGE client8a socket243 620 15125-15187
START_RANGE client8a socket244 630 15187-15250
START_RANGE client8a socket245 620 15250-15312
START_RANGE client8a socket246 630 15312-15375
START_RANGE client8a socket247 620 15375-15437
START_RANGE client8a socket248 630 15437-15500
START_RANGE client8a socket249 620 15500-15562
START_RANGE client8a socket250 630 15562-15625
START_RANGE client8a socket251 620 15625-15687
START_RANGE client8a socket252 630 15687-15750
START_RANGE client8a socket253 620 15750-15812
START_RANGE client8a socket254 630 15812-15875
START_RANGE client8a socket255 620 15875-15937
START_RANGE client8a socket256 630 15937-16000

#elif MASTER_NUM3

START_RANGE client9a socket257 620 16000-16062
START_RANGE client9a socket258 630 16062-16125
START_RANGE client9a socket259 620 16125-16187
START_RANGE client9a socket260 630 16187-16250
START_RANGE client9a socket261 620 16250-16312
START_RANGE client9a socket262 630 16312-16375
START_RANGE client9a socket263 620 16375-16437
START_RANGE client9a socket264 630 16437-16500
START_RANGE client9a socket265 620 16500-16562
START_RANGE client9a socket266 630 16562-16625
START_RANGE client9a socket267 620 16625-16687
START_RANGE client9a socket268 630 16687-16750
START_RANGE client9a socket269 620 16750-16812
START_RANGE client9a socket270 630 16812-16875
START_RANGE client9a socket271 620 16875-16937
START_RANGE client9a socket272 630 16937-17000

START_RANGE client9a socket273 620 17000-17062
START_RANGE client9a socket274 630 17062-17125
START_RANGE client9a socket275 620 17125-17187
START_RANGE client9a socket276 630 17187-17250
START_RANGE client9a socket277 620 17250-17312
START_RANGE client9a socket278 630 17312-17375
START_RANGE client9a socket279 620 17375-17437
START_RANGE client9a socket280 630 17437-17500
START_RANGE client9a socket281 620 17500-17562
START_RANGE client9a socket282 630 17562-17625
START_RANGE client9a socket283 620 17625-17687
START_RANGE client9a socket284 630 17687-17750
START_RANGE client9a socket285 620 17750-17812
START_RANGE client9a socket286 630 17812-17875
START_RANGE client9a socket287 620 17875-17937
START_RANGE client9a socket288 630 17937-18000

START_RANGE client10a socket289 620 18000-18062
START_RANGE client10a socket290 630 18062-18125
START_RANGE client10a socket291 620 18125-18187
START_RANGE client10a socket292 630 18187-18250
START_RANGE client10a socket293 620 18250-18312
START_RANGE client10a socket294 630 18312-18375
START_RANGE client10a socket295 620 18375-18437
START_RANGE client10a socket296 630 18437-18500
START_RANGE client10a socket297 620 18500-18562
START_RANGE client10a socket298 630 18562-18625
START_RANGE client10a socket299 620 18625-18687
START_RANGE client10a socket300 630 18687-18750
START_RANGE client10a socket301 620 18750-18812
START_RANGE client10a socket302 630 18812-18875
START_RANGE client10a socket303 620 18875-18937
START_RANGE client10a socket304 630 18937-19000

START_RANGE client10a socket305 620 19000-19062
START_RANGE client10a socket306 630 19062-19125
START_RANGE client10a socket307 620 19125-19187
START_RANGE client10a socket308 630 19187-19250
START_RANGE client10a socket309 620 19250-19312

START_RANGE client10a socket310 630 19312-19375
START_RANGE client10a socket311 620 19375-19437
START_RANGE client10a socket312 630 19437-19500
START_RANGE client10a socket313 620 19500-19562
START_RANGE client10a socket314 630 19562-19625
START_RANGE client10a socket315 620 19625-19687
START_RANGE client10a socket316 630 19687-19750
START_RANGE client10a socket317 620 19750-19812
START_RANGE client10a socket318 630 19812-19875
START_RANGE client10a socket319 620 19875-19937
START_RANGE client10a socket320 630 19937-20000

START_RANGE client11a socket321 620 20000-20062
START_RANGE client11a socket322 630 20062-20125
START_RANGE client11a socket323 620 20125-20187
START_RANGE client11a socket324 630 20187-20250
START_RANGE client11a socket325 620 20250-20312
START_RANGE client11a socket326 630 20312-20375
START_RANGE client11a socket327 620 20375-20437
START_RANGE client11a socket328 630 20437-20500
START_RANGE client11a socket329 620 20500-20562
START_RANGE client11a socket330 630 20562-20625
START_RANGE client11a socket331 620 20625-20687
START_RANGE client11a socket332 630 20687-20750
START_RANGE client11a socket333 620 20750-20812
START_RANGE client11a socket334 630 20812-20875
START_RANGE client11a socket335 620 20875-20937
START_RANGE client11a socket336 630 20937-21000

START_RANGE client11a socket337 620 21000-21062
START_RANGE client11a socket338 630 21062-21125
START_RANGE client11a socket339 620 21125-21187
START_RANGE client11a socket340 630 21187-21250
START_RANGE client11a socket341 620 21250-21312
START_RANGE client11a socket342 630 21312-21375
START_RANGE client11a socket343 620 21375-21437
START_RANGE client11a socket344 630 21437-21500
START_RANGE client11a socket345 620 21500-21562
START_RANGE client11a socket346 630 21562-21625
START_RANGE client11a socket347 620 21625-21687
START_RANGE client11a socket348 630 21687-21750
START_RANGE client11a socket349 620 21750-21812
START_RANGE client11a socket350 630 21812-21875
START_RANGE client11a socket351 620 21875-21937
START_RANGE client11a socket352 630 21937-22000

START_RANGE client12a socket353 620 22000-22062
START_RANGE client12a socket354 630 22062-22125
START_RANGE client12a socket355 620 22125-22187
START_RANGE client12a socket356 630 22187-22250
START_RANGE client12a socket357 620 22250-22312
START_RANGE client12a socket358 630 22312-22375
START_RANGE client12a socket359 620 22375-22437
START_RANGE client12a socket360 630 22437-22500
START_RANGE client12a socket361 620 22500-22562
START_RANGE client12a socket362 630 22562-22625
START_RANGE client12a socket363 620 22625-22687
START_RANGE client12a socket364 630 22687-22750
START_RANGE client12a socket365 620 22750-22812
START_RANGE client12a socket366 630 22812-22875
START_RANGE client12a socket367 620 22875-22937
START_RANGE client12a socket368 630 22937-23000

START_RANGE client12a socket369 620 23000-23062
START_RANGE client12a socket370 630 23062-23125
START_RANGE client12a socket371 620 23125-23187
START_RANGE client12a socket372 630 23187-23250
START_RANGE client12a socket373 620 23250-23312
START_RANGE client12a socket374 630 23312-23375
START_RANGE client12a socket375 620 23375-23437
START_RANGE client12a socket376 630 23437-23500
START_RANGE client12a socket377 620 23500-23562

START_RANGE client12a socket378 630 23562-23625
START_RANGE client12a socket379 620 23625-23687
START_RANGE client12a socket380 630 23687-23750
START_RANGE client12a socket381 620 23750-23812
START_RANGE client12a socket382 630 23812-23875
START_RANGE client12a socket383 620 23875-23937
START_RANGE client12a socket384 630 23937-24000

#elif MASTER_NUM4

START_RANGE client13a socket385 620 24000-24062
START_RANGE client13a socket386 630 24062-24125
START_RANGE client13a socket387 620 24125-24187
START_RANGE client13a socket388 630 24187-24250
START_RANGE client13a socket389 620 24250-24312
START_RANGE client13a socket390 630 24312-24375
START_RANGE client13a socket391 620 24375-24437
START_RANGE client13a socket392 630 24437-24500
START_RANGE client13a socket393 620 24500-24562
START_RANGE client13a socket394 630 24562-24625
START_RANGE client13a socket395 620 24625-24687
START_RANGE client13a socket396 630 24687-24750
START_RANGE client13a socket397 620 24750-24812
START_RANGE client13a socket398 630 24812-24875
START_RANGE client13a socket399 620 24875-24937
START_RANGE client13a socket400 630 24937-25000

START_RANGE client13a socket401 620 25000-25062
START_RANGE client13a socket402 630 25062-25125
START_RANGE client13a socket403 620 25125-25187
START_RANGE client13a socket404 630 25187-25250
START_RANGE client13a socket405 620 25250-25312
START_RANGE client13a socket406 630 25312-25375
START_RANGE client13a socket407 620 25375-25437
START_RANGE client13a socket408 630 25437-25500
START_RANGE client13a socket409 620 25500-25562
START_RANGE client13a socket410 630 25562-25625
START_RANGE client13a socket411 620 25625-25687
START_RANGE client13a socket412 630 25687-25750
START_RANGE client13a socket413 620 25750-25812
START_RANGE client13a socket414 630 25812-25875
START_RANGE client13a socket415 620 25875-25937
START_RANGE client13a socket416 630 25937-26000

START_RANGE client14a socket417 620 26000-26062
START_RANGE client14a socket418 630 26062-26125
START_RANGE client14a socket419 620 26125-26187
START_RANGE client14a socket420 630 26187-26250
START_RANGE client14a socket421 620 26250-26312
START_RANGE client14a socket422 630 26312-26375
START_RANGE client14a socket423 620 26375-26437
START_RANGE client14a socket424 630 26437-26500
START_RANGE client14a socket425 620 26500-26562
START_RANGE client14a socket426 630 26562-26625
START_RANGE client14a socket427 620 26625-26687
START_RANGE client14a socket428 630 26687-26750
START_RANGE client14a socket429 620 26750-26812
START_RANGE client14a socket430 630 26812-26875
START_RANGE client14a socket431 620 26875-26937
START_RANGE client14a socket432 630 26937-27000

START_RANGE client14a socket433 620 27000-27062
START_RANGE client14a socket434 630 27062-27125
START_RANGE client14a socket435 620 27125-27187
START_RANGE client14a socket436 630 27187-27250
START_RANGE client14a socket437 620 27250-27312
START_RANGE client14a socket438 630 27312-27375
START_RANGE client14a socket439 620 27375-27437
START_RANGE client14a socket440 630 27437-27500
START_RANGE client14a socket441 620 27500-27562
START_RANGE client14a socket442 630 27562-27625
START_RANGE client14a socket443 620 27625-27687
START_RANGE client14a socket444 630 27687-27750

START_RANGE client14a socket445 620 27750-27812
START_RANGE client14a socket446 630 27812-27875
START_RANGE client14a socket447 620 27875-27937
START_RANGE client14a socket448 630 27937-28000

START_RANGE client15a socket449 620 28000-28062
START_RANGE client15a socket450 630 28062-28125
START_RANGE client15a socket451 620 28125-28187
START_RANGE client15a socket452 630 28187-28250
START_RANGE client15a socket453 620 28250-28312
START_RANGE client15a socket454 630 28312-28375
START_RANGE client15a socket455 620 28375-28437
START_RANGE client15a socket456 630 28437-28500
START_RANGE client15a socket457 620 28500-28562
START_RANGE client15a socket458 630 28562-28625
START_RANGE client15a socket459 620 28625-28687
START_RANGE client15a socket460 630 28687-28750
START_RANGE client15a socket461 620 28750-28812
START_RANGE client15a socket462 630 28812-28875
START_RANGE client15a socket463 620 28875-28937
START_RANGE client15a socket464 630 28937-29000

START_RANGE client15a socket465 620 29000-29062
START_RANGE client15a socket466 630 29062-29125
START_RANGE client15a socket467 620 29125-29187
START_RANGE client15a socket468 630 29187-29250
START_RANGE client15a socket469 620 29250-29312
START_RANGE client15a socket470 630 29312-29375
START_RANGE client15a socket471 620 29375-29437
START_RANGE client15a socket472 630 29437-29500
START_RANGE client15a socket473 620 29500-29562
START_RANGE client15a socket474 630 29562-29625
START_RANGE client15a socket475 620 29625-29687
START_RANGE client15a socket476 630 29687-29750
START_RANGE client15a socket477 620 29750-29812
START_RANGE client15a socket478 630 29812-29875
START_RANGE client15a socket479 620 29875-29937
START_RANGE client15a socket480 630 29937-30000

START_RANGE client16a socket481 620 30000-30062
START_RANGE client16a socket482 630 30062-30125
START_RANGE client16a socket483 620 30125-30187
START_RANGE client16a socket484 630 30187-30250
START_RANGE client16a socket485 620 30250-30312
START_RANGE client16a socket486 630 30312-30375
START_RANGE client16a socket487 620 30375-30437
START_RANGE client16a socket488 630 30437-30500
START_RANGE client16a socket489 620 30500-30562
START_RANGE client16a socket490 630 30562-30625
START_RANGE client16a socket491 620 30625-30687
START_RANGE client16a socket492 630 30687-30750
START_RANGE client16a socket493 620 30750-30812
START_RANGE client16a socket494 630 30812-30875
START_RANGE client16a socket495 620 30875-30937
START_RANGE client16a socket496 630 30937-31000

START_RANGE client16a socket497 620 31000-31062
START_RANGE client16a socket498 630 31062-31125
START_RANGE client16a socket499 620 31125-31187
START_RANGE client16a socket500 630 31187-31250
START_RANGE client16a socket501 620 31250-31312
START_RANGE client16a socket502 630 31312-31375
START_RANGE client16a socket503 620 31375-31437
START_RANGE client16a socket504 630 31437-31500
START_RANGE client16a socket505 620 31500-31562
START_RANGE client16a socket506 630 31562-31625
START_RANGE client16a socket507 620 31625-31687
START_RANGE client16a socket508 630 31687-31750
START_RANGE client16a socket509 620 31750-31812
START_RANGE client16a socket510 630 31812-31875
START_RANGE client16a socket511 620 31875-31937
START_RANGE client16a socket512 630 31937-32000

```
#elif MASTER_NUM5
START_RANGE client17a socket513 620 32000-32062
START_RANGE client17a socket514 630 32062-32125
START_RANGE client17a socket515 620 32125-32187
START_RANGE client17a socket516 630 32187-32250
START_RANGE client17a socket517 620 32250-32312
START_RANGE client17a socket518 630 32312-32375
START_RANGE client17a socket519 620 32375-32437
START_RANGE client17a socket520 630 32437-32500
START_RANGE client17a socket521 620 32500-32562
START_RANGE client17a socket522 630 32562-32625
START_RANGE client17a socket523 620 32625-32687
START_RANGE client17a socket524 630 32687-32750
START_RANGE client17a socket525 620 32750-32812
START_RANGE client17a socket526 630 32812-32875
START_RANGE client17a socket527 620 32875-32937
START_RANGE client17a socket528 630 32937-33000
```

```
START_RANGE client17a socket529 620 33000-33062
START_RANGE client17a socket530 630 33062-33125
START_RANGE client17a socket531 620 33125-33187
START_RANGE client17a socket532 630 33187-33250
START_RANGE client17a socket533 620 33250-33312
START_RANGE client17a socket534 630 33312-33375
START_RANGE client17a socket535 620 33375-33437
START_RANGE client17a socket536 630 33437-33500
START_RANGE client17a socket537 620 33500-33562
START_RANGE client17a socket538 630 33562-33625
START_RANGE client17a socket539 620 33625-33687
START_RANGE client17a socket540 630 33687-33750
START_RANGE client17a socket541 620 33750-33812
START_RANGE client17a socket542 630 33812-33875
START_RANGE client17a socket543 620 33875-33937
START_RANGE client17a socket544 630 33937-34000
```

```
START_RANGE client18a socket545 620 34000-34062
START_RANGE client18a socket546 630 34062-34125
START_RANGE client18a socket547 620 34125-34187
START_RANGE client18a socket548 630 34187-34250
START_RANGE client18a socket549 620 34250-34312
START_RANGE client18a socket550 630 34312-34375
START_RANGE client18a socket551 620 34375-34437
START_RANGE client18a socket552 630 34437-34500
START_RANGE client18a socket553 620 34500-34562
START_RANGE client18a socket554 630 34562-34625
START_RANGE client18a socket555 620 34625-34687
START_RANGE client18a socket556 630 34687-34750
START_RANGE client18a socket557 620 34750-34812
START_RANGE client18a socket558 630 34812-34875
START_RANGE client18a socket559 620 34875-34937
START_RANGE client18a socket560 630 34937-35000
```

```
START_RANGE client18a socket561 620 35000-35062
START_RANGE client18a socket562 630 35062-35125
START_RANGE client18a socket563 620 35125-35187
START_RANGE client18a socket564 630 35187-35250
START_RANGE client18a socket565 620 35250-35312
START_RANGE client18a socket566 630 35312-35375
START_RANGE client18a socket567 620 35375-35437
START_RANGE client18a socket568 630 35437-35500
START_RANGE client18a socket569 620 35500-35562
START_RANGE client18a socket570 630 35562-35625
START_RANGE client18a socket571 620 35625-35687
START_RANGE client18a socket572 630 35687-35750
START_RANGE client18a socket573 620 35750-35812
START_RANGE client18a socket574 630 35812-35875
START_RANGE client18a socket575 620 35875-35937
START_RANGE client18a socket576 630 35937-36000
```

```
START_RANGE client19a socket577 620 36000-36062
START_RANGE client19a socket578 630 36062-36125
```

```
START_RANGE client19a socket579 620 36125-36187
START_RANGE client19a socket580 630 36187-36250
START_RANGE client19a socket581 620 36250-36312
START_RANGE client19a socket582 630 36312-36375
START_RANGE client19a socket583 620 36375-36437
START_RANGE client19a socket584 630 36437-36500
START_RANGE client19a socket585 620 36500-36562
START_RANGE client19a socket586 630 36562-36625
START_RANGE client19a socket587 620 36625-36687
START_RANGE client19a socket588 630 36687-36750
START_RANGE client19a socket589 620 36750-36812
START_RANGE client19a socket590 630 36812-36875
START_RANGE client19a socket591 620 36875-36937
START_RANGE client19a socket592 630 36937-37000
```

```
START_RANGE client19a socket593 620 37000-37062
START_RANGE client19a socket594 630 37062-37125
START_RANGE client19a socket595 620 37125-37187
START_RANGE client19a socket596 630 37187-37250
START_RANGE client19a socket597 620 37250-37312
START_RANGE client19a socket598 630 37312-37375
START_RANGE client19a socket599 620 37375-37437
START_RANGE client19a socket600 630 37437-37500
START_RANGE client19a socket601 620 37500-37562
START_RANGE client19a socket602 630 37562-37625
START_RANGE client19a socket603 620 37625-37687
START_RANGE client19a socket604 630 37687-37750
START_RANGE client19a socket605 620 37750-37812
START_RANGE client19a socket606 630 37812-37875
START_RANGE client19a socket607 620 37875-37937
START_RANGE client19a socket608 630 37937-38000
```

```
START_RANGE client20a socket609 620 38000-38062
START_RANGE client20a socket610 630 38062-38125
START_RANGE client20a socket611 620 38125-38187
START_RANGE client20a socket612 630 38187-38250
START_RANGE client20a socket613 620 38250-38312
START_RANGE client20a socket614 630 38312-38375
START_RANGE client20a socket615 620 38375-38437
START_RANGE client20a socket616 630 38437-38500
START_RANGE client20a socket617 620 38500-38562
START_RANGE client20a socket618 630 38562-38625
START_RANGE client20a socket619 620 38625-38687
START_RANGE client20a socket620 630 38687-38750
START_RANGE client20a socket621 620 38750-38812
START_RANGE client20a socket622 630 38812-38875
START_RANGE client20a socket623 620 38875-38937
START_RANGE client20a socket624 630 38937-39000
```

```
START_RANGE client20a socket625 620 39000-39062
START_RANGE client20a socket626 630 39062-39125
START_RANGE client20a socket627 620 39125-39187
START_RANGE client20a socket628 630 39187-39250
START_RANGE client20a socket629 620 39250-39312
START_RANGE client20a socket630 630 39312-39375
START_RANGE client20a socket631 620 39375-39437
START_RANGE client20a socket632 630 39437-39500
START_RANGE client20a socket633 620 39500-39562
START_RANGE client20a socket634 630 39562-39625
START_RANGE client20a socket635 620 39625-39687
START_RANGE client20a socket636 630 39687-39750
START_RANGE client20a socket637 620 39750-39812
START_RANGE client20a socket638 630 39812-39875
START_RANGE client20a socket639 620 39875-39937
START_RANGE client20a socket640 630 39937-40000
```

```
#elif MASTER_NUM6
START_RANGE client21a socket641 620 40000-40062
START_RANGE client21a socket642 630 40062-40125
START_RANGE client21a socket643 620 40125-40187
START_RANGE client21a socket644 630 40187-40250
START_RANGE client21a socket645 620 40250-40312
```

START_RANGE client21a socket646 630 40312-40375
START_RANGE client21a socket647 620 40375-40437
START_RANGE client21a socket648 630 40437-40500
START_RANGE client21a socket649 620 40500-40562
START_RANGE client21a socket650 630 40562-40625
START_RANGE client21a socket651 620 40625-40687
START_RANGE client21a socket652 630 40687-40750
START_RANGE client21a socket653 620 40750-40812
START_RANGE client21a socket654 630 40812-40875
START_RANGE client21a socket655 620 40875-40937
START_RANGE client21a socket656 630 40937-41000

START_RANGE client21a socket657 620 41000-41062
START_RANGE client21a socket658 630 41062-41125
START_RANGE client21a socket659 620 41125-41187
START_RANGE client21a socket660 630 41187-41250
START_RANGE client21a socket661 620 41250-41312
START_RANGE client21a socket662 630 41312-41375
START_RANGE client21a socket663 620 41375-41437
START_RANGE client21a socket664 630 41437-41500
START_RANGE client21a socket665 620 41500-41562
START_RANGE client21a socket666 630 41562-41625
START_RANGE client21a socket667 620 41625-41687
START_RANGE client21a socket668 630 41687-41750
START_RANGE client21a socket669 620 41750-41812
START_RANGE client21a socket670 630 41812-41875
START_RANGE client21a socket671 620 41875-41937
START_RANGE client21a socket672 630 41937-42000

START_RANGE client22a socket673 620 42000-42062
START_RANGE client22a socket674 630 42062-42125
START_RANGE client22a socket675 620 42125-42187
START_RANGE client22a socket676 630 42187-42250
START_RANGE client22a socket677 620 42250-42312
START_RANGE client22a socket678 630 42312-42375
START_RANGE client22a socket679 620 42375-42437
START_RANGE client22a socket680 630 42437-42500
START_RANGE client22a socket681 620 42500-42562
START_RANGE client22a socket682 630 42562-42625
START_RANGE client22a socket683 620 42625-42687
START_RANGE client22a socket684 630 42687-42750
START_RANGE client22a socket685 620 42750-42812
START_RANGE client22a socket686 630 42812-42875
START_RANGE client22a socket687 620 42875-42937
START_RANGE client22a socket688 630 42937-43000

START_RANGE client22a socket689 620 43000-43062
START_RANGE client22a socket690 630 43062-43125
START_RANGE client22a socket691 620 43125-43187
START_RANGE client22a socket692 630 43187-43250
START_RANGE client22a socket693 620 43250-43312
START_RANGE client22a socket694 630 43312-43375
START_RANGE client22a socket695 620 43375-43437
START_RANGE client22a socket696 630 43437-43500
START_RANGE client22a socket697 620 43500-43562
START_RANGE client22a socket698 630 43562-43625
START_RANGE client22a socket699 620 43625-43687
START_RANGE client22a socket700 630 43687-43750
START_RANGE client22a socket701 620 43750-43812
START_RANGE client22a socket702 630 43812-43875
START_RANGE client22a socket703 620 43875-43937
START_RANGE client22a socket704 630 43937-44000

START_RANGE client23a socket705 620 44000-44062
START_RANGE client23a socket706 630 44062-44125
START_RANGE client23a socket707 620 44125-44187
START_RANGE client23a socket708 630 44187-44250
START_RANGE client23a socket709 620 44250-44312
START_RANGE client23a socket710 630 44312-44375
START_RANGE client23a socket711 620 44375-44437
START_RANGE client23a socket712 630 44437-44500
START_RANGE client23a socket713 620 44500-44562

START_RANGE client23a socket714 630 44562-44625
START_RANGE client23a socket715 620 44625-44687
START_RANGE client23a socket716 630 44687-44750
START_RANGE client23a socket717 620 44750-44812
START_RANGE client23a socket718 630 44812-44875
START_RANGE client23a socket719 620 44875-44937
START_RANGE client23a socket720 630 44937-45000

START_RANGE client23a socket721 620 45000-45062
START_RANGE client23a socket722 630 45062-45125
START_RANGE client23a socket723 620 45125-45187
START_RANGE client23a socket724 630 45187-45250
START_RANGE client23a socket725 620 45250-45312
START_RANGE client23a socket726 630 45312-45375
START_RANGE client23a socket727 620 45375-45437
START_RANGE client23a socket728 630 45437-45500
START_RANGE client23a socket729 620 45500-45562
START_RANGE client23a socket730 630 45562-45625
START_RANGE client23a socket731 620 45625-45687
START_RANGE client23a socket732 630 45687-45750
START_RANGE client23a socket733 620 45750-45812
START_RANGE client23a socket734 630 45812-45875
START_RANGE client23a socket735 620 45875-45937
START_RANGE client23a socket736 630 45937-46000

START_RANGE client24a socket737 620 46000-46062
START_RANGE client24a socket738 630 46062-46125
START_RANGE client24a socket739 620 46125-46187
START_RANGE client24a socket740 630 46187-46250
START_RANGE client24a socket741 620 46250-46312
START_RANGE client24a socket742 630 46312-46375
START_RANGE client24a socket743 620 46375-46437
START_RANGE client24a socket744 630 46437-46500
START_RANGE client24a socket745 620 46500-46562
START_RANGE client24a socket746 630 46562-46625
START_RANGE client24a socket747 620 46625-46687
START_RANGE client24a socket748 630 46687-46750
START_RANGE client24a socket749 620 46750-46812
START_RANGE client24a socket750 630 46812-46875
START_RANGE client24a socket751 620 46875-46937
START_RANGE client24a socket752 630 46937-47000

START_RANGE client24a socket753 620 47000-47062
START_RANGE client24a socket754 630 47062-47125
START_RANGE client24a socket755 620 47125-47187
START_RANGE client24a socket756 630 47187-47250
START_RANGE client24a socket757 620 47250-47312
START_RANGE client24a socket758 630 47312-47375
START_RANGE client24a socket759 620 47375-47437
START_RANGE client24a socket760 630 47437-47500
START_RANGE client24a socket761 620 47500-47562
START_RANGE client24a socket762 630 47562-47625
START_RANGE client24a socket763 620 47625-47687
START_RANGE client24a socket764 630 47687-47750
START_RANGE client24a socket765 620 47750-47812
START_RANGE client24a socket766 630 47812-47875
START_RANGE client24a socket767 620 47875-47937
START_RANGE client24a socket768 630 47937-48000

#elif MASTER_NUM7

START_RANGE client25a socket769 620 48000-48062
START_RANGE client25a socket770 630 48062-48125
START_RANGE client25a socket771 620 48125-48187
START_RANGE client25a socket772 630 48187-48250
START_RANGE client25a socket773 620 48250-48312
START_RANGE client25a socket774 630 48312-48375
START_RANGE client25a socket775 620 48375-48437
START_RANGE client25a socket776 630 48437-48500
START_RANGE client25a socket777 620 48500-48562
START_RANGE client25a socket778 630 48562-48625
START_RANGE client25a socket779 620 48625-48687
START_RANGE client25a socket780 630 48687-48750

START_RANGE client25a socket781 620 48750-48812
START_RANGE client25a socket782 630 48812-48875
START_RANGE client25a socket783 620 48875-48937
START_RANGE client25a socket784 630 48937-49000

START_RANGE client25a socket785 620 49000-49062
START_RANGE client25a socket786 630 49062-49125
START_RANGE client25a socket787 620 49125-49187
START_RANGE client25a socket788 630 49187-49250
START_RANGE client25a socket789 620 49250-49312
START_RANGE client25a socket790 630 49312-49375
START_RANGE client25a socket791 620 49375-49437
START_RANGE client25a socket792 630 49437-49500
START_RANGE client25a socket793 620 49500-49562
START_RANGE client25a socket794 630 49562-49625
START_RANGE client25a socket795 620 49625-49687
START_RANGE client25a socket796 630 49687-49750
START_RANGE client25a socket797 620 49750-49812
START_RANGE client25a socket798 630 49812-49875
START_RANGE client25a socket799 620 49875-49937
START_RANGE client25a socket800 630 49937-50000

START_RANGE client26a socket801 620 50000-50062
START_RANGE client26a socket802 630 50062-50125
START_RANGE client26a socket803 620 50125-50187
START_RANGE client26a socket804 630 50187-50250
START_RANGE client26a socket805 620 50250-50312
START_RANGE client26a socket806 630 50312-50375
START_RANGE client26a socket807 620 50375-50437
START_RANGE client26a socket808 630 50437-50500
START_RANGE client26a socket809 620 50500-50562
START_RANGE client26a socket810 630 50562-50625
START_RANGE client26a socket811 620 50625-50687
START_RANGE client26a socket812 630 50687-50750
START_RANGE client26a socket813 620 50750-50812
START_RANGE client26a socket814 630 50812-50875
START_RANGE client26a socket815 620 50875-50937
START_RANGE client26a socket816 630 50937-51000

START_RANGE client26a socket817 620 51000-51062
START_RANGE client26a socket818 630 51062-51125
START_RANGE client26a socket819 620 51125-51187
START_RANGE client26a socket820 630 51187-51250
START_RANGE client26a socket821 620 51250-51312
START_RANGE client26a socket822 630 51312-51375
START_RANGE client26a socket823 620 51375-51437
START_RANGE client26a socket824 630 51437-51500
START_RANGE client26a socket825 620 51500-51562
START_RANGE client26a socket826 630 51562-51625
START_RANGE client26a socket827 620 51625-51687
START_RANGE client26a socket828 630 51687-51750
START_RANGE client26a socket829 620 51750-51812
START_RANGE client26a socket830 630 51812-51875
START_RANGE client26a socket831 620 51875-51937
START_RANGE client26a socket832 630 51937-52000

START_RANGE client27a socket833 620 52000-52062
START_RANGE client27a socket834 630 52062-52125
START_RANGE client27a socket835 620 52125-52187
START_RANGE client27a socket836 630 52187-52250
START_RANGE client27a socket837 620 52250-52312
START_RANGE client27a socket838 630 52312-52375
START_RANGE client27a socket839 620 52375-52437
START_RANGE client27a socket840 630 52437-52500
START_RANGE client27a socket841 620 52500-52562
START_RANGE client27a socket842 630 52562-52625
START_RANGE client27a socket843 620 52625-52687
START_RANGE client27a socket844 630 52687-52750
START_RANGE client27a socket845 620 52750-52812
START_RANGE client27a socket846 630 52812-52875
START_RANGE client27a socket847 620 52875-52937
START_RANGE client27a socket848 630 52937-53000

START_RANGE client27a socket849 620 53000-53062
START_RANGE client27a socket850 630 53062-53125
START_RANGE client27a socket851 620 53125-53187
START_RANGE client27a socket852 630 53187-53250
START_RANGE client27a socket853 620 53250-53312
START_RANGE client27a socket854 630 53312-53375
START_RANGE client27a socket855 620 53375-53437
START_RANGE client27a socket856 630 53437-53500
START_RANGE client27a socket857 620 53500-53562
START_RANGE client27a socket858 630 53562-53625
START_RANGE client27a socket859 620 53625-53687
START_RANGE client27a socket860 630 53687-53750
START_RANGE client27a socket861 620 53750-53812
START_RANGE client27a socket862 630 53812-53875
START_RANGE client27a socket863 620 53875-53937
START_RANGE client27a socket864 630 53937-54000

START_RANGE client28a socket865 620 54000-54062
START_RANGE client28a socket866 630 54062-54125
START_RANGE client28a socket867 620 54125-54187
START_RANGE client28a socket868 630 54187-54250
START_RANGE client28a socket869 620 54250-54312
START_RANGE client28a socket870 630 54312-54375
START_RANGE client28a socket871 620 54375-54437
START_RANGE client28a socket872 630 54437-54500
START_RANGE client28a socket873 620 54500-54562
START_RANGE client28a socket874 630 54562-54625
START_RANGE client28a socket875 620 54625-54687
START_RANGE client28a socket876 630 54687-54750
START_RANGE client28a socket877 620 54750-54812
START_RANGE client28a socket878 630 54812-54875
START_RANGE client28a socket879 620 54875-54937
START_RANGE client28a socket880 630 54937-55000

START_RANGE client28a socket881 620 55000-55062
START_RANGE client28a socket882 630 55062-55125
START_RANGE client28a socket883 620 55125-55187
START_RANGE client28a socket884 630 55187-55250
START_RANGE client28a socket885 620 55250-55312
START_RANGE client28a socket886 630 55312-55375
START_RANGE client28a socket887 620 55375-55437
START_RANGE client28a socket888 630 55437-55500
START_RANGE client28a socket889 620 55500-55562
START_RANGE client28a socket890 630 55562-55625
START_RANGE client28a socket891 620 55625-55687
START_RANGE client28a socket892 630 55687-55750
START_RANGE client28a socket893 620 55750-55812
START_RANGE client28a socket894 630 55812-55875
START_RANGE client28a socket895 620 55875-55937
START_RANGE client28a socket896 630 55937-56000

#elif MASTER_NUM8

START_RANGE client29a socket897 620 56000-56062
START_RANGE client29a socket898 630 56062-56125
START_RANGE client29a socket899 620 56125-56187
START_RANGE client29a socket900 630 56187-56250
START_RANGE client29a socket901 620 56250-56312
START_RANGE client29a socket902 630 56312-56375
START_RANGE client29a socket903 620 56375-56437
START_RANGE client29a socket904 630 56437-56500
START_RANGE client29a socket905 620 56500-56562
START_RANGE client29a socket906 630 56562-56625
START_RANGE client29a socket907 620 56625-56687
START_RANGE client29a socket908 630 56687-56750
START_RANGE client29a socket909 620 56750-56812
START_RANGE client29a socket910 630 56812-56875
START_RANGE client29a socket911 620 56875-56937
START_RANGE client29a socket912 630 56937-57000

START_RANGE client29a socket913 620 57000-57062
START_RANGE client29a socket914 630 57062-57125

```
START_RANGE client29a socket915 620 57125-57187
START_RANGE client29a socket916 630 57187-57250
START_RANGE client29a socket917 620 57250-57312
START_RANGE client29a socket918 630 57312-57375
START_RANGE client29a socket919 620 57375-57437
START_RANGE client29a socket920 630 57437-57500
START_RANGE client29a socket921 620 57500-57562
START_RANGE client29a socket922 630 57562-57625
START_RANGE client29a socket923 620 57625-57687
START_RANGE client29a socket924 630 57687-57750
START_RANGE client29a socket925 620 57750-57812
START_RANGE client29a socket926 630 57812-57875
START_RANGE client29a socket927 620 57875-57937
START_RANGE client29a socket928 630 57937-58000
```

```
START_RANGE client30a socket929 620 58000-58062
START_RANGE client30a socket930 630 58062-58125
START_RANGE client30a socket931 620 58125-58187
START_RANGE client30a socket932 630 58187-58250
START_RANGE client30a socket933 620 58250-58312
START_RANGE client30a socket934 630 58312-58375
START_RANGE client30a socket935 620 58375-58437
START_RANGE client30a socket936 630 58437-58500
START_RANGE client30a socket937 620 58500-58562
START_RANGE client30a socket938 630 58562-58625
START_RANGE client30a socket939 620 58625-58687
START_RANGE client30a socket940 630 58687-58750
START_RANGE client30a socket941 620 58750-58812
START_RANGE client30a socket942 630 58812-58875
START_RANGE client30a socket943 620 58875-58937
START_RANGE client30a socket944 630 58937-59000
```

```
START_RANGE client30a socket945 620 59000-59062
START_RANGE client30a socket946 630 59062-59125
START_RANGE client30a socket947 620 59125-59187
START_RANGE client30a socket948 630 59187-59250
START_RANGE client30a socket949 620 59250-59312
START_RANGE client30a socket950 630 59312-59375
START_RANGE client30a socket951 620 59375-59437
START_RANGE client30a socket952 630 59437-59500
START_RANGE client30a socket953 620 59500-59562
START_RANGE client30a socket954 630 59562-59625
START_RANGE client30a socket955 620 59625-59687
START_RANGE client30a socket956 630 59687-59750
START_RANGE client30a socket957 620 59750-59812
START_RANGE client30a socket958 630 59812-59875
START_RANGE client30a socket959 620 59875-59937
START_RANGE client30a socket960 630 59937-60000
```

```
START_RANGE client31a socket961 620 60000-60062
START_RANGE client31a socket962 630 60062-60125
START_RANGE client31a socket963 620 60125-60187
START_RANGE client31a socket964 630 60187-60250
START_RANGE client31a socket965 620 60250-60312
START_RANGE client31a socket966 630 60312-60375
START_RANGE client31a socket967 620 60375-60437
START_RANGE client31a socket968 630 60437-60500
START_RANGE client31a socket969 620 60500-60562
START_RANGE client31a socket970 630 60562-60625
START_RANGE client31a socket971 620 60625-60687
START_RANGE client31a socket972 630 60687-60750
START_RANGE client31a socket973 620 60750-60812
START_RANGE client31a socket974 630 60812-60875
START_RANGE client31a socket975 620 60875-60937
START_RANGE client31a socket976 630 60937-61000
```

```
START_RANGE client31a socket977 620 61000-61062
START_RANGE client31a socket978 630 61062-61125
START_RANGE client31a socket979 620 61125-61187
START_RANGE client31a socket980 630 61187-61250
START_RANGE client31a socket981 620 61250-61312
START_RANGE client31a socket982 630 61312-61375
```

```
START_RANGE client31a socket983 620 61375-61437
START_RANGE client31a socket984 630 61437-61500
START_RANGE client31a socket985 620 61500-61562
START_RANGE client31a socket986 630 61562-61625
START_RANGE client31a socket987 620 61625-61687
START_RANGE client31a socket988 630 61687-61750
START_RANGE client31a socket989 620 61750-61812
START_RANGE client31a socket990 630 61812-61875
START_RANGE client31a socket991 620 61875-61937
START_RANGE client31a socket992 630 61937-62000
```

```
START_RANGE client32a socket993 620 62000-62062
START_RANGE client32a socket994 630 62062-62125
START_RANGE client32a socket995 620 62125-62187
START_RANGE client32a socket996 630 62187-62250
START_RANGE client32a socket997 620 62250-62312
START_RANGE client32a socket998 630 62312-62375
START_RANGE client32a socket999 620 62375-62437
START_RANGE client32a socket1000 630 62437-62500
START_RANGE client32a socket1001 620 62500-62562
START_RANGE client32a socket1002 630 62562-62625
START_RANGE client32a socket1003 620 62625-62687
START_RANGE client32a socket1004 630 62687-62750
START_RANGE client32a socket1005 620 62750-62812
START_RANGE client32a socket1006 630 62812-62875
START_RANGE client32a socket1007 620 62875-62937
START_RANGE client32a socket1008 630 62937-63000
```

```
START_RANGE client32a socket1009 620 63000-63062
START_RANGE client32a socket1010 630 63062-63125
START_RANGE client32a socket1011 620 63125-63187
START_RANGE client32a socket1012 630 63187-63250
START_RANGE client32a socket1013 620 63250-63312
START_RANGE client32a socket1014 630 63312-63375
START_RANGE client32a socket1015 620 63375-63437
START_RANGE client32a socket1016 630 63437-63500
START_RANGE client32a socket1017 620 63500-63562
START_RANGE client32a socket1018 630 63562-63625
START_RANGE client32a socket1019 620 63625-63687
START_RANGE client32a socket1020 630 63687-63750
START_RANGE client32a socket1021 620 63750-63812
START_RANGE client32a socket1022 630 63812-63875
START_RANGE client32a socket1023 620 63875-63937
START_RANGE client32a socket1024 630 63937-64000
```

```
#elif MASTER_NUM9
#endif
/*-----*/
#define TES_FLAG_TRACE 0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000020
#define TES_FLAG_LOCAL_LOG 0x00000400
#define TES_FLAG_LOCAL_TRACE 0x00000800
#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="

#define PAYMENT_RESULTS_CID_URL "/tpcc/tpcc.html?00=pymt&01=%d&03=%d&04=%d&05=&06=
%d&07=%d&08=%d.%02.2d"
#define PAYMENT_RESULTS_CLAST_URL "/tpcc/tpcc.html?00=pymt&01=%d&03=%d&04=&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=&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 strcasecmp(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 %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 WHSEARRAYDBG
    outfboundwarn=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;

    iprint (IPRINT_TRACE, "Keystroke times = ");
    for (i = 0 ; i < num_timestamps; i++) {
        iprint (IPRINT_TRACE, "%d ", times[i]);
    }
    iprint (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 (!strcasecmp (ptr, "pause")) {
        shmglobal->test_state = 1;
    } else if (!strcasecmp (ptr, "warmup")) {
        shmglobal->test_state = 2;
    } else if (!strcasecmp (ptr, "notest")) {
        shmglobal->test_state = 0;
    } else if (!strcasecmp (ptr, "login_max_load?")) {
        iprint (IPRINT_WARNING, "Current LOGIN_MAX_LOAD =
%d\n", login_max_load);
    } else if (!strcasecmp (command, "login_max_load=", 15)) {
        login_max_load = atoi(command+15);
        iprint (IPRINT_WARNING, "Set LOGIN_MAX_LOAD = %d\n",
login_max_load);
    } else if (!strcasecmp (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 (!strcasecmp (ptr, Status_Names[i], len)) {
                    status_needs_refresh = found = 1;
                    current_status = i;
                    return RTE_OK;
                }
            }
            iprint (IPRINT_WARNING, "Unknown type to display: %s\n",
ptr);
        }
    } else {
        iprint (IPRINT_WARNING, "Unknown Command: %s\n",
command);
        return RTE_ERROR;
    }
    return RTE_OK;
}

int transaction_process () {
    return RTE_OK;
}

int user_begin() {
    return RTE_OK;
}

/*
void user_make_header(char *buffer) {
    int i;
    struct user_data_header *data = (struct user_data_header
*)buffer;
}
*/

user_slave.C
/*****
*****/
/* user_slave.C                      Audit: 05/30/96 */
/*****

static char *rcsid="$Id: user_slave.C,v 1.1 1999/02/22 06:31:06
channui Exp $";

/*****

```

```

/**          TPCC FILE FOR ALL USERS          ***/
/*****
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/time.h>
#include "rte_slave.h"
#include "user_tpcc.h"

/* This MUST match the corresponding one in client's inout.h file! */
#define TRIGGER "\021"
#define NOSLEEP
// Increased EXPECT_TIMEOUT from 600000 - oz 10/20/97
#define EXPECT_TIMEOUT 600000
#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, 10000 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_T
ITLE_LEN);
        break;
    case '2':

```

```

        rc =
expect_html(PYMT_RESULTS_TITLE,timeout,PYMT_RESULTS_TI
TLE_LEN);
        break;
    case '3':
        rc =
expect_html(ORDS_RESULTS_TITLE,timeout,ORDS_RESULTS_T
ITLE_LEN);
        break;
    case '4':
        rc =
expect_html(DLVY_RESULTS_TITLE,timeout,DLVY_RESULTS_TI
TLE_LEN);
        break;
    case '5':
        rc =
expect_html(STOK_RESULTS_TITLE,timeout,STOK_RESULTS_TI
TLE_LEN);
        break;
    default:
        rc = ERROR;
}

if(rc == ERROR)
{
    iprint (IPRINT_ERROR, "Slave %d:Failed to receive %s
result page\n**Request:-->%s<--\n",
        shmentry->num, data->menu,data->form_request);
    return (ERROR);
}

// if (expect_html(TRIGGER, timeout) == ERROR) {
//iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s
response\n",
//    shmentry->num, data->menu);
//    return (ERROR);
//}

#ifdef NOSLEEP
    usleep(shmglobal->emulex_response[data-
>type]*1000000.0+0.9);
#endif

// Look for errors and set our think time (THINK)
transaction_mark(WHERE_NOW);
if (expect_buffer_search("ERROR:",6)
{
    FILE *fd;
    if ((fd = fopen("/u/rte/error.txt","a"))!=NULL)
    {
        fprintf(fd,"Slave %d: %s found '%s'\n%s\n",
            shmentry->num, data->menu,
"ERROR:",expect_buffer_return());
        fclose(fd);
    }
    data->invalid = 1;
    iprint (IPRINT_ERROR, "Slave %d: %s found '%s'\n%s\n",
        shmentry->num, data->menu,
"ERROR:",expect_buffer_return());
    // Very dangerous, keep going rather than exiting...
    //return RTE_ERROR;
    // Check for consecutive errors and if there are more than
    // 4 of them exit - allow for transient errors to make
    // tuning and testing easier -oz
    // In either case the transaction is marked as invalid and
    // will be reported as an error by the analyze program.
    if (consecutive_errs++ > 4)
        return RTE_ERROR;
}
else

```

```

{
    consecutive_errs = 0;
}

echo_trace ("Thinking");
transaction_sleep_set(neg_exp_4(shmglobal->think[data-
>type])*1000.0);
//iprint(IPRINT_TRACE, "< generic_transaction finish\n");
return (RTE_OK);
}

/*****
***      Delivery Transaction      ***
*****/
int
Delivery(char *host,int terminal)
{
    static struct delivery_struct delivery, delivery_new;
    int      rc;
    char      *ptr;

    gen_tran_t  tran;

    tran.invalid = 0;
    tran.data = &delivery;
    tran.len = sizeof(delivery);
    tran.keywait = 2;
    tran.type = DELIVERY;
    tran.menu = "4";

    char dlvy_url[128];
    char form_buffer[256];
    char results_buffer[256];

    tran.form_request = form_buffer;
    tran.results_request = results_buffer;

    //create dlvy form request
    sprintf(dlvy_url,DELIVERY_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,dlvy_url,host);

    // Set up all data for new transactions
    delivery_new.carrier = uniform(1, 10); // carrier # 1 to 10

    //create dlvy results request
    sprintf(dlvy_url,DELIVERY_RESULTS_URL,terminal,delivery_n
ew.carrier);
    sprintf(results_buffer,GET_REQUEST,dlvy_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    delivery = delivery_new;
    delivery.invalid = tran.invalid;

    //iprint(IPRINT_TRACE,"dlvy txn finished, rc:%d
tran.invalid:%d\n",rc,delivery.invalid);

    return (rc);
}

/*****
***      New Order Transaction      ***
*****/
int NewOrder(char *host,int terminal)
{
    static struct neword_struct neword, neword_new;
    int      i, rc, whses, low_whse=1;
    char nord_form_url[128];
    char form_buffer[512];

```

```

char nord_results_url[2048];
char results_buffer[4096];

char      *ptr;
char      *ptr2;
const char      *err_ptr;

gen_tran_t  tran;

tran.invalid = 0;
tran.data = &neword;
tran.len = sizeof(neword);
tran.keywait = 18;
tran.type = NEWORDER;
tran.menu = "1";

tran.form_request = form_buffer;
tran.results_request = results_buffer;

neword_new.rollback=0;

/** SECTION TO DETERMINE ROLLBACK TRANSACTION FOR
1% OF NEW ORDERS **/
neword_new.did = uniform(1, 10); // district
number
neword_new.cid = NURand(1023, 1, 3000, CUSTC); //
customer # 1 to 3000
neword_new.nloop = uniform(5, 15); //
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
sprintf(nord_form_url,NEW_ORDER_FORM_URL,terminal);

//create get form request
sprintf(form_buffer,GET_REQUEST,nord_form_url,host);

```

```

//create new order results url
char itemString[1024];
ptr2=itemString;
short item_cmd_start = ITEM_CMD_ID_START;
for (i = 0; i < neword_new.nloop; i++)
{
    ptr2 += sprintf(ptr2, NEW_ORDER_ITEM,
                    item_cmd_start++,
                    neword_new.item[i].olswid);

    ptr2 += sprintf(ptr2,NEW_ORDER_ITEM,
                    item_cmd_start++,
                    neword_new.item[i].oliid);

    ptr2 += sprintf(ptr2, NEW_ORDER_ITEM,
                    item_cmd_start++,
                    neword_new.item[i].olquantity);
}
//seal up url w/ empty items
for (i = item_cmd_start;i <= ITEM_CMD_ID_END; i++)
{
    ptr2 += sprintf(ptr2,NEW_ORDER_EMPTY_ITEM,i);
}

sprintf(nord_results_url,NEW_ORDER_RESULTS_URL,termin
al,
neword_new.did,neword_new.cid,
                                itemString);

//create get results request
sprintf(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");
    neword.oid = -1;
}
else
{
    neword.oid = atoi(err_ptr+14);
    // iprint(IPRINT_ERROR,"New order order
id:%d\n",neword.oid);
}

// This is really not useful since we aren't going to be sending
individual
// keystrokes anymore

```

```

if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
    log_data(RTE_ITEM_KEYSTROKE_TIMES,
            keystroke_length*sizeof(int),keystroke_times);
}

//iprint(IPRINT_TRACE,"nord txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}

/*****
***      Order Status Transaction      ***
*****/
int OrderStatus(char *host,int terminal) {
    static struct ordstat_struct ordstat, ordstat_new;
    //char    buffer[2048];
    int      rc;
    char     *ptr;
    gen_tran_t  tran;

    tran.invalid = 0;
    tran.data   = &ordstat;
    tran.len    = sizeof(ordstat);
    tran.keywait = 2;
    tran.type   = ORDSTAT;
    tran.menu   = "3";
    //tran.request = buffer;

    //Joe N.
    char ords_url[256];
    char form_buffer[512];
    char results_buffer[2048];

    tran.results_request = results_buffer;
    tran.form_request = form_buffer;

    //create order status form request
    sprintf(ords_url,ORDER_STATUS_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,ords_url,host);

    // Set up all data for new transactions
    ordstat_new.did = uniform(1, 10); /* district number 1 to 10 */
    if (uniform(1, 100) <= 60)
    {
        /* for 60% of transactions */
        char *tmp = getname();
        strcpy(ordstat_new.clast, tmp); /* by customer last
name */
        if (ordstat_new.clast[0] < 'A' || ordstat_new.clast[0] > 'Z')
        {
            iprint (IPRINT_ERROR,
                    "ASSERTION: OrderStatus getname() returns invalid
name! '%s\n",
                    ordstat_new.clast);
            return RTE_ERROR;
        }
        ordstat_new.byname = 1;
        ordstat_new.cid = 0;
    }
    else
    {
        ordstat_new.cid = NURand(1023, 1, 3000, CUSTC); /*
cust. # 1 to 3000 */
        ordstat_new.byname = 0;
        ordstat_new.clast[0] = (char) NULL;
    }

    //iprint(IPRINT_TRACE,"Order status fields,w_id:%d d_id:%d
n_d_id:%d c_id:%d\n", data->Warehouse,data-
>District,ordstat_new.did,ordstat_new.cid);

```



```

//create order status url request
if (ordstat_new.byname)

    sprintf(ords_url,ORDER_STATUS_RESULTS_CLAST_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;

    //create pymt form url
    sprintf(pymt_url,PAYMENT_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,pymt_url,host);

    payment_new.did = uniform(1, 10); /* district number 1 to 10 */
    if (uniform(1, 100) <= 60) /* for 60% of transactions */
        strncpy(payment_new.clast, getname(), 17); // by customer
last name
    if (payment_new.clast[0] < 'A' || payment_new.clast[0] > 'Z') {
        iprint (IPRINT_ERROR,
            "ASSERTION: payment_new getname() returns invalid
name! '%s'\n",
            payment_new.clast);
        return RTE_ERROR;
    }
}

```

```

    payment_new.byname = 1;
    payment_new.cid = 0;
} else {
    payment_new.cid = NURand(1023, 1, 3000, CUSTC); /* cust.
# 1 to 3000 */
    payment_new.byname = 0;
    payment_new.clast[0] = (char) NULL;
}

whses = shmglobal->max_warehouses;

if (whses < 2 || uniform(1, 100) <= 85) /* for 85 % of transactions
*/
    payment_new.cwid = WHSEID;
    payment_new.cdoid = payment_new.did;
    payment_new.remote = 0;
} else { /* for 15 % of transactions */
    payment_new.cwid = (long) uniform((long)low_whse, (long)
whses-1);
    if (payment_new.cwid >= WHSEID)
        payment_new.cwid++;

    payment_new.remote = 1;
    payment_new.cdoid = uniform(1, 10); /* district 1 to 10 */
}

dollars = uniform(1, 5000); /* dollar amt = 1 to 5000 */
if (dollars == 5000)
    cents = 0;
else
    cents = uniform(0, 99);

    payment_new.amount = ((double) dollars) + ((double) cents) /
100.0;

//create payment results url
if (payment_new.byname)

    sprintf(pymt_url,PAYMENT_RESULTS_CLAST_URL,terminal,
payment_new.did,payment_new.clast,payment_new.cwid,
payment_new.cdoid,dollars,cents);
    else

        sprintf(pymt_url,PAYMENT_RESULTS_CID_URL,terminal,
payment_new.did,payment_new.cid,payment_new.cwid,
payment_new.cdoid,dollars,cents);
        sprintf(results_buffer,GET_REQUEST,pymt_url,host);

// Go do the transaction

rc = generic_transaction(&tran,host);
payment = payment_new;
payment.invalid = tran.invalid;

//iprint(IPRINT_TRACE,"pymt txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);

return (rc);
}

/*****
/**      Stock Level Transaction      ***/
/*****
int
StockLevel(char *host,int terminal)

```

```

{
static struct stocklev_struct stocklevel, stocklevel_new;

int rc;
char *ptr;
gen_tran_t tran;

tran.invalid = 0;
tran.data = &stocklevel;
tran.len = sizeof(stocklevel);
tran.keywait = 2;
tran.type = STOCKLEV;
tran.menu = "5";

char stok_url[128];
char form_buffer[256];
char results_buffer[2048];

tran.results_request = results_buffer;
tran.form_request = form_buffer;

//create stok form url
sprintf(stok_url,STOCK_FORM_URL,terminal);
sprintf(form_buffer,GET_REQUEST,stok_url,host);

stocklevel_new.invalid = 0;
stocklevel_new.threshold = uniform(10, 20);/* uniform no. between
10 and
* 20 */

//create stok results url
sprintf(stok_url,STOCK_RESULTS_URL,terminal,stocklevel_n
ew.threshold);
sprintf(results_buffer,GET_REQUEST,stok_url,host);

// Go do the transaction
rc = generic_transaction(&tran,host);
stocklevel = stocklevel_new;
stocklevel.invalid = tran.invalid;

//iprint(IPRINT_TRACE,"stok txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}

/*****
*** MAIN() ****
*****/
int
user_transaction(char *host,void *data,int terminal)
{
UserLocal *localdata = (UserLocal *)data;

char logout[32];
double ntask;
int resp;
static int task = 0;

if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME)
{
int rc;
/* Wait for specified period of time */
sleep (shmglobal->keystroke_sleep);
/* Quit after one transaction */
shm->lock(shmentry->pid);
shmentry->flags |= TES_FLAG_DIE;

shm->unlock(shmentry->pid);
rc = NewOrder(host,terminal);
iprint (IPRINT_INFO, "Slave %d: Keystroke timing setting
die flag\n", shmentry->num);

```

```

return rc;
}

#if 1
switch (shmglobal->test_state)
{
case 0: // Normal
break;

case 1: // pause
sleep (1);
return RTE_OK;

case 2: // warmup
switch(task++)
{
case 0: return Delivery(host,terminal);
case 1: return OrderStatus(host,terminal);
case 2: return Payment(host,terminal);
case 3: return StockLevel(host,terminal);
case 4: task = 0; return NewOrder(host,terminal);
return NewOrder(host,terminal);
}
}
/*****
*** CHOOSE ONE OF THE TRANSACTIONS
*****/
/*****
ntask = (double) uniform(0.0, 100.0);
if (ntask <= shmglobal->chances[DELIVERY])
{
return Delivery(host,terminal);
//return NewOrder(host,terminal);
}
ntask -= shmglobal->chances[DELIVERY];
if (ntask <= shmglobal->chances[ORDSTAT])
{
return OrderStatus(host,terminal);
}
ntask -= shmglobal->chances[ORDSTAT];
if (ntask <= shmglobal->chances[PAYMENT])
{
return Payment(host,terminal);
}
ntask -= shmglobal->chances[PAYMENT];
if (ntask <= shmglobal->chances[STOCKLEV])
{
return StockLevel(host,terminal);
}
}
return NewOrder(host,terminal);

#else
{
int deck[100], count=-1, i, size=1, tmp;
// lock deck
if (count < 0) {
// deck is empty fill it up
count = 0;
for (i = 0; i < 43 * size; i++) {
deck[count++] = Payment;
}
for (i = 0; i < 4 * size; i++) {
deck[count++] = StockLevel;
}
for (i = 0; i < 4 * size; i++) {
deck[count++] = OrderStatus;
}
for (i = 0; i < 4 * size; i++) {
deck[count++] = Delivery;
}
for (; count < 100 * size; i++) {
deck[count++] = NewOrder;
}
}

```

```

// randomize the deck
for (i = 0; i < 100 * size; i++) {
    int tmp;
    int pick = uniform(i+1, 100);
    tmp = deck[i];
    deck[i] = deck[pick];
    deck[pick] = tmp;
}
tmp = deck[count-];
// unlock deck

switch(tmp) {
case Delivery: return Delivery(host,terminal);
case OrderStatus: return OrderStatus(host,terminal);
case Payment: return Payment(host,terminal);
case StockLevel: return StockLevel(host,terminal);
case NewOrder: return NewOrder(host,terminal);
}

/*
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_cstux_inf.");
    return (ERROR);
} else
    return (RTE_OK);
#endif
} /* end of Main */

int user_parameter_change(void) {
    #if 0
    int i;
    iprint(IPRINT_TRACE, "Slave %d: total_users = %d\n", shmentry->num);
    iprint(IPRINT_TRACE, "Slave %d: chances = ", shmentry->num);
    for (i = 0; i < MAX_TRAN_TYPE; i++)
        iprint(IPRINT_TRACE, "%6.2f ", shmglobal->chances[i]);
    iprint(IPRINT_TRACE, "\nSlave %d: think = ", shmentry->num);
    for (i = 0; i < MAX_TRAN_TYPE; i++)
        iprint(IPRINT_TRACE, "%6.2f ", shmglobal->think[i]);
    #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, "Terminal 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 ORDDSTAT 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: Third Party Pricing Information



800.750.4239

SHOPPING CART

[▶ Your Saved Carts](#)
 [▶ Save This Cart](#)
 [▶ Edit Saved Carts](#)
 [▶ Send To An Associate](#)

[Continue to Checkout ▶▶](#)

Quantity	Product	CDW	Usually Ships	Price	Ext. Price
<input type="text" value="1"/>	Microsoft MS Visual Studio .NET Professional 2003 - complete package	528577	Same Day	\$1,016.44	\$1,016.44
<input type="text" value="3"/>	NETGEAR FS750AT 48-Port 10/100 Switch	500969	Same Day	\$486.00	\$1,458.00
				Sub-Total	\$2,474.44

Click to remove an item from your cart

[Update](#)
 [Clear Cart](#)

[Continue to Checkout ▶▶](#)

[Continue Shopping](#) | [Go to CDW.com Homepage](#)

Related Top Sellers For: [NETGEAR FS750AT 48-Port 10/100 Switch](#)

[VIEW MORE ▶](#)

▶ **Accessories**



Netgear AG711 Gigabit Module

\$248.87

Shipping Calc:

Enter a postal code to quickly estimate shipping cost.

QuickCart:

Enter a CDW **part number** to quickly add it to your cart.

*** Sample: CDW Part #**

Usually Ships:	Same Day
CDW Part:	XXXXXX
Mfg. Part:	XXXXXX-XXXXXX
UNSPSC:	XXXXXXXX