TPC Benchmark™ C Full Disclosure Report Using Sun Microsystems Enterprise 450/480 Mhz Server and Fujitsu SymfoWARE Server for Workgroup 2.0.1 RDBMS

Sun
microsystems

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C™ test conducted on the Sun Enterprise 450/480 Mhz Server system, running Fujitsu SymfoWARE Server for Workgroup 2.0.1 RDBMS and BEA Systems, Inc. Tuxedo 6.3.

TPC Benchmark C Metrics

The standard TPC Benchmark™C metrics, tpmC (transactions per minute), price per tpmC (five year capital cost per measured tpmC), and the availability date are reported as required by the benchmark specification.

Executive Summary Statements

Pages v-vii contain the executive summary of the benchmark result for the Sun Microsystems Enterprise Server 450 .

First Printing

Sun Microsystems, Inc believes that the information in this document is accurate as of its publication date. The information in this document is subject to change without notice. Sun Microsystems, Inc assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect prices in effect on August 21,2000. However, Sun Microsystems, Inc provides no warranty on the pricing information in this document.

The performance information in this document is for guidance only. System performance is highly dependent on many factors including system hardware, system and user software, and user application characteristics. Customer applications must be carefully evaluated before estimating performance. Sun Microsystems Computer Company does not warrant or represent that a user can or will achieve a similar performance expressed in tpmC or normalized price/performance ($/tpmC). No warranty on system performance or price/performance is expressed or implied in this document.

🔆 *Sun* ∞ **FUJITSU**	**Sun Enterprise 450 C/S w/8 Front-Ends**	TPC-C 3.5
		Report Date: **August 21, 2000**

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$541,817	**25,375.03** **tpmC**	**$21.35**	**October 31, 2000**

Processors	Database Manager	Operating System	Other Software	Number of Users
4 * 480MHz UltraSPARC II	**SymfoWARE Server for Workgroup 2.0.1**	**Solaris 8**	**BEA Tuxedo 6.3**	**21120**

21120 User Connections to 8Clients

Sun Enterprise 450 Server
4 * 480Mhz UltraSPARC CPU
4 GB RAM
16 x 18.2GB Drives

32 * Sun StorEdge MultiPack

8 * Sun Ultra 10 Model 440Server
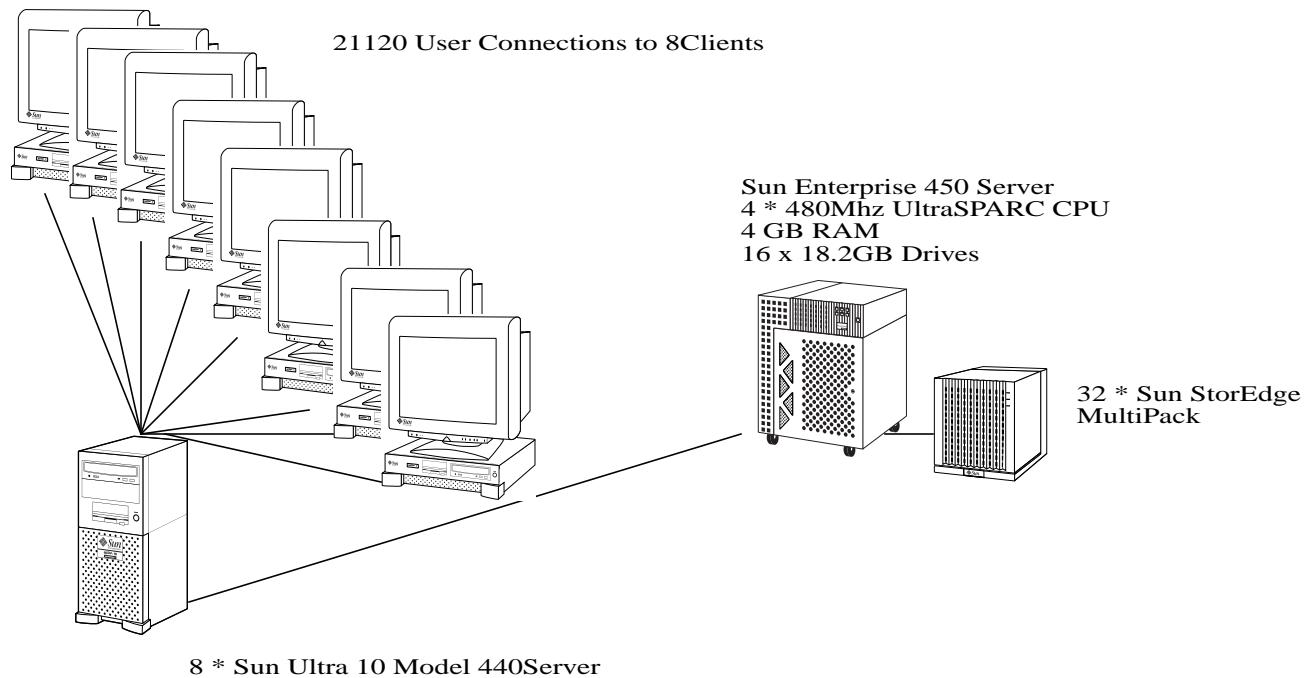
Configuration

	Server System	Front End Systems
Database Nodes:	1 Sun Enterprise E450 Server	8 * Ultra 10 Model 440
Processors	4* 480 MHz UltraSPARC II	1 * 440MHz UltraSPARC II each
Cache memory	32KB (D+I), 8MB external	32KB (D+I), 2MB external, each
Main memory	4 GB	1 GB each
Disk controllers	8x IntraServer PCI/SCSI Controllers 2x PCI UltraSCSI Controllers	1 * SCSI-2 each
Disk Drives	205* 18.2 GB 10K RPM SCSI	1 * 9.1GB 10K RPM SCSI
Total Disk Storage	3731 GB	9.1 GB each
10 BaseT Hub	None	2904 * 9-Port Hubs
100 Base T Hub	1 x 8-port Hub	None

Pricing Summary

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5 Yr. Maint.
Server Hardware						
Sun Enterprise E450 Server Base	A25-CA	2	9,984	1	9,984	13,993
480MHz/8MB UltraSPARC II	X2248A	2	6,156	4	24,622	
512-Mbyte Memory Expansion	7005A	2	3,080	8	24,640	
Expansion kit for Internal Storage	6601A	2	999	2	1,998	
IntraServer PCI/SCSI Controller*	ITI-6200U2-S	6	463	10	4,625	
nternal 18.2-Gbyte, 10K RPM Disk	5237A	2	915	21	19,219	
36.4-Gbyte StorEdge MultiPack	SG-XDSK020C-36G	2	3106	2	6,213	
109.2-Gbyte StorEdge MultiPack	SG-XDSK060C-109G	2	6089.6	30	182,688	
4mm DDS-3 Tape Drive	6286A	2	986	1	986	
Wyse55 General Purpose Terminal	901237-01	2	500	1	500	
Cable-68/68pin SCSI W/PWR CRD	X3856A	2	40	16	634	
Cable, SCSI3/VHDCI	530-2453	2	108	16	1,728	
North American/Asia PWR CRD	X311L	2	0	16	0	
Server Hardware Subtotal					**277,836**	**13,993**

*Qty includes 2 for spares

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5 Yr. Maint.
Server Software						
Solaris Server Software	SOLMS-26ZW9999	1	100	1	100	
SPARC Compiler C/C++ 5.0	WCCIS-500-T999	1	995	1	995	1,080
Fujitsu COBOL Standard Edition		3	2500	1	2,500	2,500
SymfoWARE Server for Workgroup 2.0		3	34790	1	34,790	17,395
Server Software Subtotal					**38,385**	**20,975**

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5 Yr. Maint.
Client Hardware						
Ultra 10 Server Model 440	A22UKC1Z9S-B512CP	2	4,528	8	36,221	34,445
North American/Asia PWR CRD	X311L	2	0	8	0	
512MB Memory for Ultra 10	7039A	2	1,580	8	12,637	
PCI QFE Card	1034A	2	1,292	8	10,339	
Color Monitor	x7143A	2	286	8	2,288	
Client Hardware Subtotal			**7,686**		**61,485**	**34,445**

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5 Yr. Maint.
Client Software						
BEA Tuxedo CFS 6.3		4	3,000	8	24,000	19,200
Client Software Subtotal					**24,000**	**19,200**

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5 Yr. Maint.
User Connectivity						
8-Port 10/100Mbps Ethernet Hub	Z222667	5	114	3	342	
9-port 10Mbps Ethernet Hub	Z222666	5	23	2904	66,792	
User Connectivity Subtotal					**67,134**	
Sun Enterprise Services discounts						**(15,636)**
			Total		468,840	72,977
			5Yr. cost		541,817	
			tpmC Rating		25,375	
			$/tpmC		$21.35	

Service for all Sun products is from Sun Microsystems, Inc.

Notes:

1. Sun Microsystems Inc. 2. CAT Technology Inc 3. Fujitsu 4. BEA Systems, Inc. 5. Software House Int 6. IntraServer

Audited by: Bradley J. Askins, Infosizing Inc.

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 purchase 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 the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

Numerical Quantity Summary

MQTH, Computed Maximum Qualified Throughput   **= 25,375.03 tpmC**

% throughput difference, reported & reproducibility runs   = < 0.1%

Response Times (in secs)	90th Percentile	Average	Maximum
Menu	0.50	0.246	0.373
New-Order	2.00	.910	42.822
Payment	2.20	1.081	43.808
Order-Status	1.80	.960	14.304
Delivery(interactive)	0.310	0.285	4.958
Delivery(deferred)	3.00	.951	7.000
Stock-level	2.00	.845	6.954

Transaction Mix, in percent of total transactions

New-Order	44.83%
Payment	43.04%
Order-Status	4.04%
Delivery	4.02%
Stock-level	4.07%

Keying/Think Times (in secs)	Average.	Min.	Maximum
New-Order	18.02/12.19	18.01/0	18.14/122.00
Payment	3.02/12.20	3.01/0	3.18/122.00
Order-Status	2.02/10.21	2.01/0	2.08/102.50
Delivery	2.02/5.18	2.01/0	2.08/52.00
Stock-level	2.02/5.20	2.01/0	2.09/52.00

Test Duration	
Ramp-up time	25 minutes
Measurement Interval	30 minutes
Number of checkpoints	1
Checkpoint Interval	30 minutes
Number of transactions (all types) completed in measurement interval	1698058

Contents

Preface

This report documents the compliance of the Sun Microsystems TPC Benchmark ™C testing on the Enterprise 450 Server running Fujitsu SymfoWARE Server for Workgroup 2.0.1 with the *TPC Benchmark* ^{TM}*C Standard Revision 3.5.*

These tests were run using the Fujitsu SymfoWARE Server for Workgroup 2.0.1 running with Solaris 8 on the Enterprise 450 Server and BEA Tuxedo 6.3 on the Ultra 10 Model 440 clients.

Document Structure

The *TPC Benchmark* ™*C Full Disclosure Report* is organized as follows:

- The main body of the document lists each item in Clause **8** of the TPC Benchmark ™C Standard and explains how each specification is satisfied.

- Appendix A contains the application source code that implements the transactions and forms modules.

- Appendix B contains the code used to create and load the database.

- Appendix C contains the configuration information for the operating system, the RDBMS and Tuxedo.

- Appendix D contains the 180-day space calculations.

- Appendix E contains the code used to generate transactions and measure response times.

- Appendix F contains the screen layouts of all the forms.

- Appendix G contains the price quotes.

Additional Copies

To request additional copies of this report, write to the following address:

Shanley P.R.
777 N First Street, Suite 600
San Jose, CA 95112-6311

(408) 295-8894

FAX (408) 295-2613

Enterprise 450 TPC Benchmark ™C Full Disclosure

Introduction

The *TPC Benchmark ™C Standard Specification* requires test sponsors to publish, and make available to the public, a full disclosure report for the results to be considered compliant with the Standard. The required contents of the full disclosure report are specified in Clause 8.

This report is intended to satisfy the Standard's requirement for full disclosure. It documents the compliance of the benchmark tests reported in the *TPC Benchmark ™C* results for the Sun Microsystems Enterprise 450 Server running Fujitsu SymfoWARE Server for Workgroup 2.0.1.

In the *Standard Specification*, the main headings in Clause 8 are keyed to the other clauses. The headings in this report use the same sequence, so that they correspond to the titles or subjects referred to in Clause 8.

Each section in this report begins with the text of the corresponding item from Clause 8 of the *Standard Specification*, printed in italic type. The plain type text that follows explains how the tests comply with the TPC C™ Benchmark requirement. In sections where Clause 8 requires extensive listings, the section refers to the appropriate appendix at the end of this report.

1- General Items

1.1 Application Code and Definition Statements

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 and B contains the application source code that implements the transactions and forms modules.

1.2 Sponsor

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

This benchmark test was sponsored by Sun Microsystems, Inc. and Fujitsu, Inc.

1.3 Parameter Settings

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

- *Database tuning options*

- *Recovery/commit options*

- *Consistency/locking options*

- *Operating system and application configuration parameters*

- *Compilation and linkage options and run-time optimizations used to create/install applications, OS, and/or databases.*

This requirement can be satisfied by providing a full list of all parameters and options.

Appendix C contains all the required parameter settings.

1.4 Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The measured configuration was the same as the priced configuration, with the exception that 194 18.2GB disks were priced in place of 194 9.1GB disks. Figure 1 is a diagram of the configuration.

Configuration Items for the Enterprise 450

For the configuration, the server machine was a Sun Enterprise 450 which consisted of the following:

- Sun Enterprise E450 Server Base

- 4 UltraSPARC-II 480 MHz Processors with 8MB External Cache each

- 4 GB of main memory

- 2 Expansion kits for Internal Storage. (Each include a PCI UltraSCSI controller.)

- 11 Internal 9.1-GByte, 10K RPM disks (Priced as 18.2 GB)

- 5 Internal 18.2-GByte, 10K RPM disks

- 8 IntraServer PCI UltraSCSI controllers

- 28 Sun StorEdge 54.6-GByte MultiPacks (6 x 9 GB SCSI disks in each)*

- 1 Sun StorEdge 109.2-GByte MultiPacks (6 x 18 GB SCSI disks in each)

- 3 Sun StorEdge 18.2-GByte MultiPacks (5 x 9 GB SCSI disk in each.)*

- Internal CD-ROM

- 4mm DDS-3 Backup Tape Device

*These drives were substituted with 18.2 GB when priced

The eight client machines were Ultra 10S UltraSPARC-II 440 MHz system. Each contained:

- One UltraSPARC-II 440 MHz Processor.

- 1024 MB of Main Memory.

- One Internal SCSI-2 controller.

- One Internal 9 GB SCSI disk.

- Internal CD-ROM.

- Quad FastEthernet Controller

The benchmark configuration used a Remote Terminal Emulator (RTE) to emulate TPC-C user sessions. The driver systems were directly connected through ethernet to the clients which emulated the database client sessions.

Driver Nodes   Client Nodes   Server Node

8 * Ethernet 10 baseT

4 * 480 Mhz UltraSPARC CPU
11 x 9.1 GB Drives *
5 x 18.2 GB Drives

8 * Ethernet 100 baseT

8 * Ultra 10 Model 440
- 1 UltraSPARC-II 440 MHz Processor
- 1 GB Memory

28 * Sun StorEdge
MultiPack *
- 6 x 9GB Each

Sun StorEdge
MultiPack
- 6 x 18 GB

3* Sun StorEdge
MultiPack*
- 5 x 9 GB

* All 9.1 GB drives were substituted with 18.2 GB
drives in priced configuration

Figure 1: The Sun Enterprise 450 Benchmark Configuration

2 - Clause 1 Related Items

2.1 Table Definitions

Listing must be provided for all table definition statements and all other statements used to set up the database.

Appendix B describes the programs that define, create, and populate a Fujitsu SymfoWARE database for TPC-C testing.

2.2 Physical Organization of Database

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

Appendix B discloses the organization of tables and indices on the disks.

2.3 Insert and 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 restrictions in the SUT database 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.

All insert and delete functions were verified and fully operational during the entire benchmark.

2.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark (see Clause 1.6), any such partitioning must be disclosed.

All tables were horizontally partitioned except for Items. Each table was horizontally partitioned following the w-id values given below:

Table 1: DSI (Data Structure Instance)

Table	w-id
Warehouse	88
District	88
Customer	22
History	22
Orders	22
New order	22
Order line	11
Stock	44

2.5 Table Replication

Replication of tables, if used, must be disclosed (see Clause 1.4.6).

No tables were replicated in this implementation.

2.6 Table Attributes

Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance (see Clause 1.4.7).

No additional or duplicate attributes were added to any of the tables.

3 - Clause 2 Related Items

3.1 Random Number Generation

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

The Random Number Generator used was the one that appeared in the article titled "Random Number Generators: Good Ones Are Hard To Find" in the communications of the ACM - October 1988, Volume 31, Number 10. The properties of this random

number generator are well-known and are documented in the article as producing a uniformly distributed pseudo-random sequence. To generate a random number, the driver programs first use a seed based on the host address, current time and the process-id of the respective session. This guarantees that each emulated user on all the RTE machines is mathematically independent of others.

3.2 Input/Output Screen Layouts

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

The screen layouts are shown in Appendix F.

3.3 Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained.

The terminal attributes were verified by the auditor manually exercising each specification during the onsite audit portion of this benchmark.

3.4 Presentation Manager or Intelligent Terminal

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

The TPC-C forms module was implemented using the capabilities of an xterm terminal emulator.

3.5 Transaction Statistics

Table 1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 requires.

Table 1: Transaction Statistics

Transaction Type	Statistics	Percentage
New Order	Home warehouse	98.995
	Remote warehouse	1.005
	Rolled back transactions	.992
	Average items per order	10.001

Table 1: Transaction Statistics

Transaction Type	Statistics	Percentage
Payment	Home warehouse	85.064
	Remote warehouse	14.936
	Non-primary key access	60.034
Order Status	Non-primary key access	60.048
Delivery	Skipped transactions	0.00
Transaction Mix	New order	44.83
	Payment	43.04
	Order status	4.04
	Delivery	4.02
	Stock level	4.07

3.6 Queueing Mechanism

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

Delivery transactions were submitted to servers using the same Tuxedo call mechanism that other transactions used. The only difference was that the call was asynchronous - i.e., control returned to the client process immediately and the deferred delivery completed asynchronously.

4 - Clause 3 Related Items

4.1 Transaction System Properties (ACID)

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

The TPC Benchmark C Standard Specification defines a set of transaction processing system properties that a system under test (SUT) must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation, and Durability (ACID).

This section defines each of these properties, describes the steps taken to ensure that they were present during the test and describes a series of tests done to demonstrate compliance with the standard.

4.2 Atomicity

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

4.2.1 Completed Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

A row was randomly selected from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

4.2.2 Aborted Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) 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.

 A row was randomly selected from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

4.3 Consistency

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

The benchmark specification requires explicit demonstration of the following four consistency conditions:

* The sum of the district balances in a warehouse is equal to the warehouse balance;

* For each district, the next order id minus one is equal to the maximum order id in the ORDER table and equal to the maximum new order id in the NEW-ORDER table;

* For each district, the maximum order id minus minimum order id in the ORDER table plus one equals the number of rows in the NEW-ORDER table for that district;

* For each district, the sum of the order line counts in the ORDER table equals the number of rows in the ORDER-LINE table for that district;

These consistency conditions were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

A performance run was completed including a full 30 minutes of steady state and checkpoints.

The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

4.4 Isolation

Isolation can be defined in terms of phenomena that can occur during the execution of concurrent transactions. These phenomena are P0 ("Dirty Write"), P1 ("Dirty Read"), P2 ("Non-repeatable Read") and P3 ("Phantom"). The table in Clause 3.4.1 of

the TPC-C specifications defines the isolation requirements which must be met by the TPC-C transactions. Sufficient conditions must be enabled at either the system or application level to ensure the required isolation is maintained.

Isolation tests one through nine were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate the required isolation had been met.

For Isolation test seven, case A was followed.

4.5 Durability

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

List of single failures:

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

Instantaneous interruption (system crash/system hang) in processing which requires system reboot to recover.

Failure of all or part of memory (loss of contents).

4.5.1 Durable Media Failure

To demonstrate recovery from a permanent failure of durable medial containing the SymfoWARE recovery log data and TPC-C tables, the following steps were executed on a database of 2,112 warehouses:

1. The database was backed up to extra disks.

2. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.

3. The RTEs were started with 21,120 users.

4. The test was allowed to run for a minimum of 5 minutes.

5. One of the log disks was powered off by removing it from the cabinet. Since the log was mirrored, the transactions continued to run without interruption.

6. The test was allowed to run for another 5 minutes and a disk failure was caused by removing a disk from the disk cabinet.

7. The RTEs were shut down.

8. A new disk was inserted into the disk cabinet and the data disk was reformatted to simulate a complete loss of data.

9. SymfoWARE was restarted.

10. Data from the backup disk was copied to the new disk and SymfoWARE used the transaction logs to roll forward the recovery data from committed transactions.

11. Step 2 was repeated and the difference between the first and second counts noted.

12. The success file was used to determine the number of NEW_ORDERS successfully returned to the RTEs.

13. The counts in step 11 and 12 were compared, and the results verified that all committed transactions were successfully recovered.

14. Data from the success file was used to query the database to demonstrate that successful transactions had corresponding rows in the ORDER table and that rolled back transactions did not.

4.5.2 Instantaneous Interruption and Loss of Memory

Because loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was executed on a fully scaled database of 2,112 warehouses under a full load of 21,120 users. The following steps were executed:

1. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.

2. The RTE was started with 21,120 users.

3. The test was allowed to run for a minimum of 30 minutes.

4. A checkpoint was enforced.

5. The test was allowed to run for another minute.

6. The primary power to the processor was shutdown.

7. The RTE was shutdown.

8. Power was restored and the system performed an automatic recovery.

9. SymfoWARE was restarted and performed an automatic recovery.

10. Step 1 was repeated and the difference between the first and second countswas noted.

11. The success file was used to determine the number of NEW-ORDERS successfully returned to the RTE.

12. The counts in step 10 and 11 were compared and the results verified that all committed transactions had been successfully recovered.

13. Data from the success file was used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table, and rolled back transactions did not.

5 - Clause 4 Related Items

5.1 Initial Cardinality of Tables

The Cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2) the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

Table 2: Initial Cardinality of Tables

Table	Occurrences
Warehouse	2,112
District	21,120
Customer	63,360,000
History	63,360,000
Orders	63,360,000
New order	19,008,000

Table 2: Initial Cardinality of Tables

Table	Occurrences
Order line	633,605,888
Stock	211,200,000
Item	100,000

5.2 Database Layout

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

The distribution of database tables over the 192 disks, and the logs across 10 disks of the system is the same distribution of the tested system, 180 day storage growth requirements are made with the substitution of 18 GB drives for some of the 9 GB drives. Figure 1 shows the configuration of the system disks.

5.2.1 Database Layout of Benchmark System.

Table 3: Disk Layout

Device Name	No. of Devices/ Structures	Physical Disks
Log Devices/Mirrors (8 hr)	7	10 disks
warehouse	24	24 disks
district	24	24 disks
customer	96	192 disks
history	96	192 disks
order	96	192 disks
new-order	96	192 disks
order-line	192	192 disks
stock	48	192 disks
item	1	10 disks

The data was striped across a total of 192 disks, 6 of which were 18GB.. 11 of these were either internal to the E450 or located in 32 Sun StorEdge MultiPacks. An additional 2x 9GB disks were used for the Operating System, swap disks and Fujitsu binaries.

The logs were located on 10 physical disks, 4 of which were 18 GB.

5.3 Type of Database

A statement must be provided that describes:

1. The data model implemented by the DBMS used (e.g., relational, network hierarchical).

2. The database interface (e.g., embedded, call level) and access language (e.g., SQL, DL/1, COBOL read/write) used to implement the TPC-C transactions. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.

SymfoWARE is a relational database management system.The interface used was SymfoWARE stored procedures embedded in C code.

5.4 Mapping of Database

The mapping of database partitions/replications must be explicitly described.

The database, with the exception of the Item table, was horizontally partitioned. This partitioning is fully described in Section1.4..

5.5 180 Day Space Computation

Details of the 180 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 (see Clause 4.2.3).

The 180 day space computation is shown in Appendix D.

The archive log grows at the rate of 5.5564KB per New-Order transaction, which was measured from the steady state. The 8 hours log space was 64.54GB at the measured rate and 67.44GB of log space was prepared for the measurement.

For dynamic tables the following steps were followed:

1. The number of rows and number of used blocks were counted on a freshly loaded database.

2. The number of rows was divided by the number of blocks, giving rows per   block.

3. The number of rows inserted in 8 hours was estimated equal to tpmC for HISTORY and ORDER, and ten times tpmC for ORDERLINE.

4. The number of rows in step 3 was divided by the number derived in step 2.

5. The number in step 4 was added to the number of used blocks from step 1.

6. The database was queried to show the space allocated exceeded the number in step 5.

6 - Clause 5 Related Items

6.1 Measured tpmC

Measured tpmC must be reported.

The measured tpmC was *25375.03*

6.2 Response Times

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

Table 4: Response Times

Type	Average	Maximum	90% percentile
New-Order	0.910	42.822	2.000
Payment	1.081	43.808	2.200
Order-Status	0.960	14.304	1.800
Interactive Delivery	0.285	4.958	0.310
Deferred Delivery	0.951	7.00	3.00
Stock-Level	0.845	6.954	2.00
Menu	0.246	0.373	0.500

6.3 Keying and Think Times

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

Table 5: Keying Times

Type	Average	Minimum	Maximum
New-Order	18.02	18.01	18.14
Payment	3.02	3.01	3.18
Order-Status	2.02	2.01	2.08
Interactive Delivery	2.02	2.01	2.08
Stock-Level	2.02	2.01	2.09

Table 6: Think Times

Type	Average	Minimum	Maximum
New-Order	12.19	0.00	122.0
Payment	12.20	0.00	122.0
Order-Status	10.21	0.00	102.5
Interactive Delivery	5.18	0.00	52.00
Stock-Level	5.20	0.00	52.00

6.4 Response Time Frequency Distribution Curves

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type.
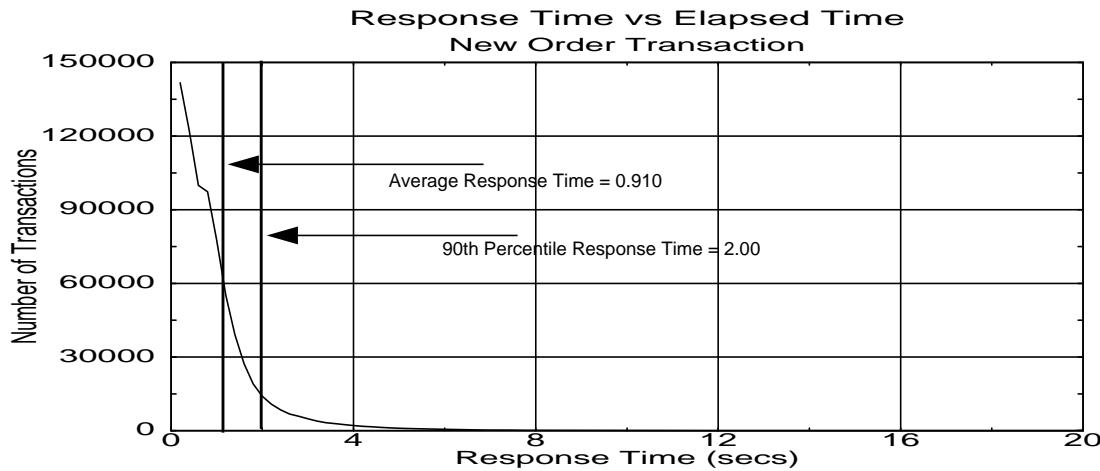
Figure 5: New Order Response Time Distribution
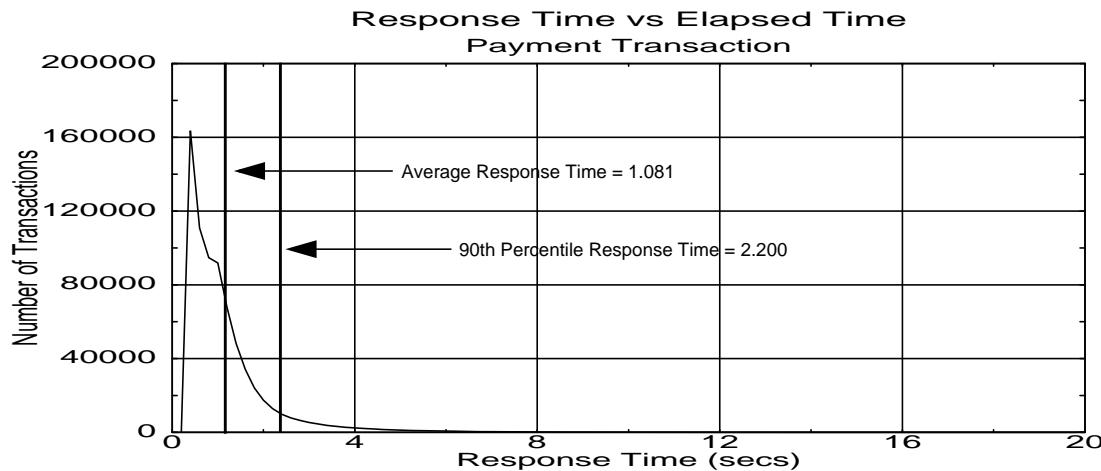
Figure 6: Payment Response Time Distribution
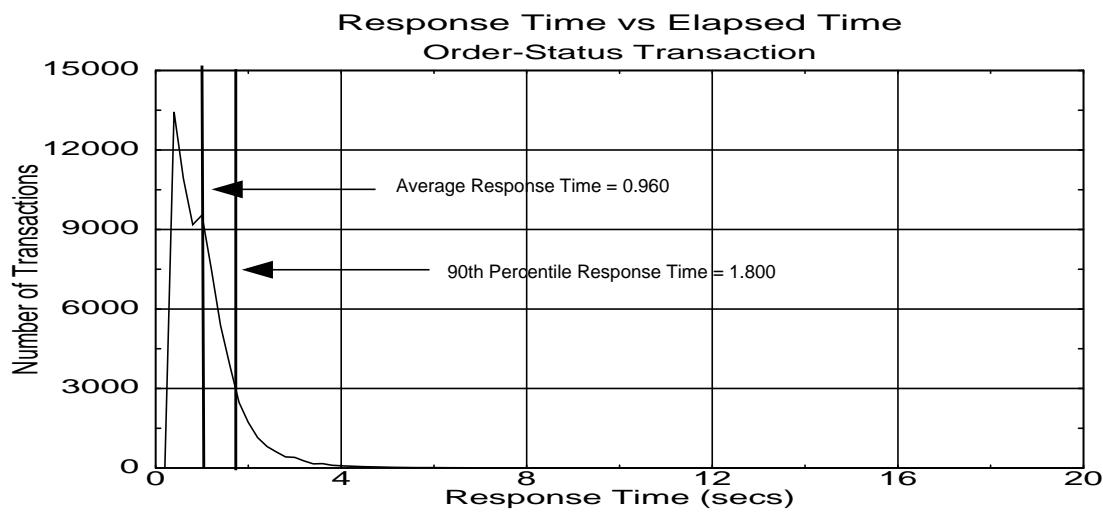
Figure 7: Order Status Response Time Distribution
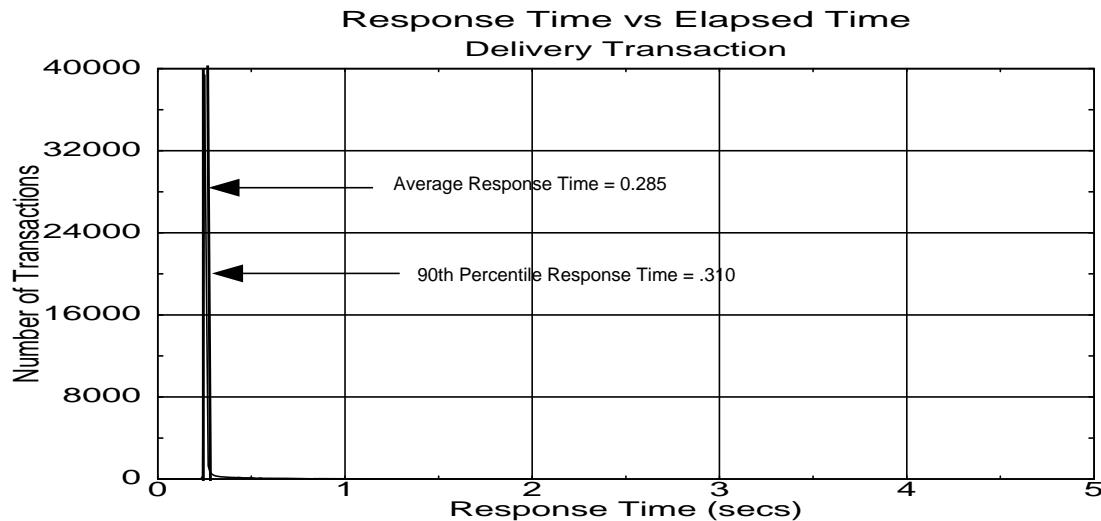
Figure 8: Delivery Response Time Distribution
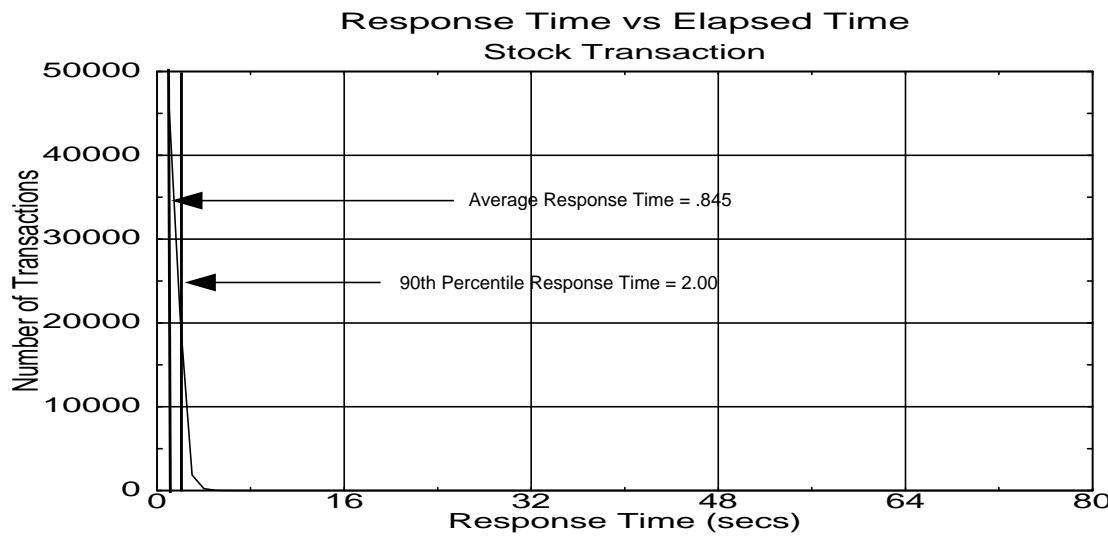
Figure 9: Stock Level Response Time Distribution

6.5 Response time versus throughput

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New Order transaction.
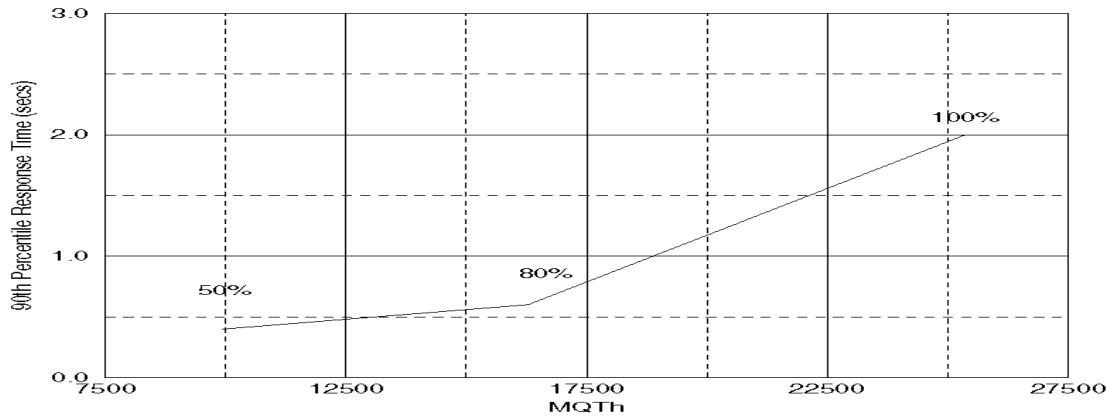
Figure 10: Response Time versus Throughput

TPC Benchmark ™C Full Disclosure-August 2000

6.6 Think Time distribution curves

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type.

Think Time vs Elapsed Time
New Order Transaction
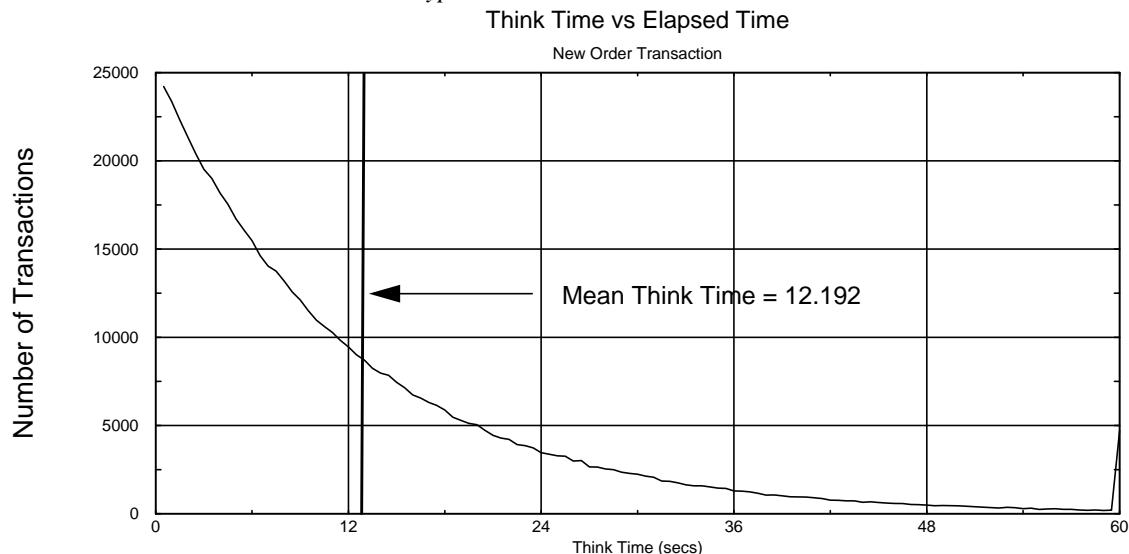
Figure 11: New Order Think Time Distribution

Think Time vs Elapsed Time
Payment Transaction
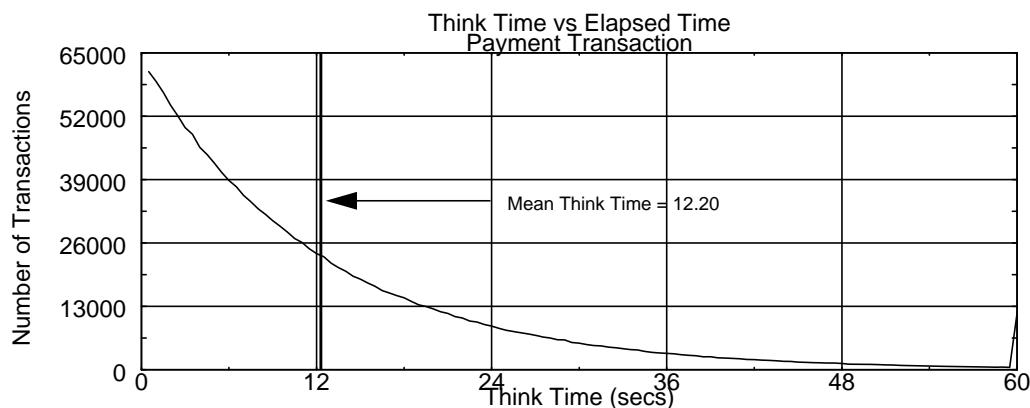
Figure 12: Payment Think Time Distribution
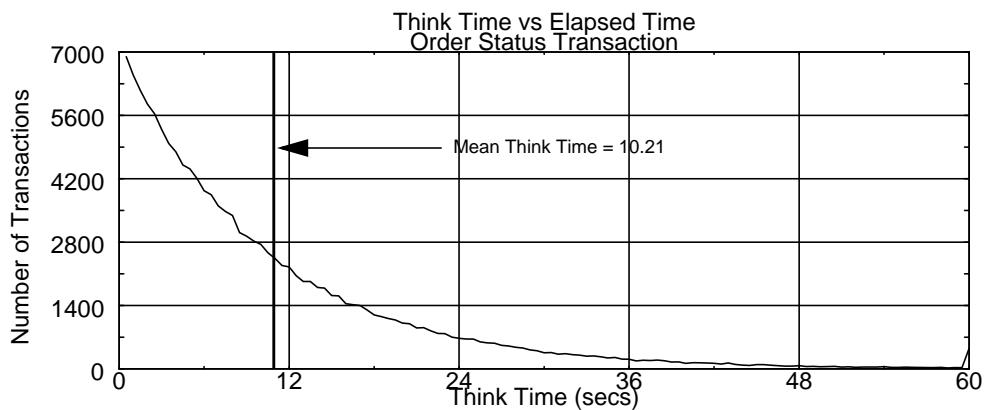
Figure 13: Order Status Think Time Distribution
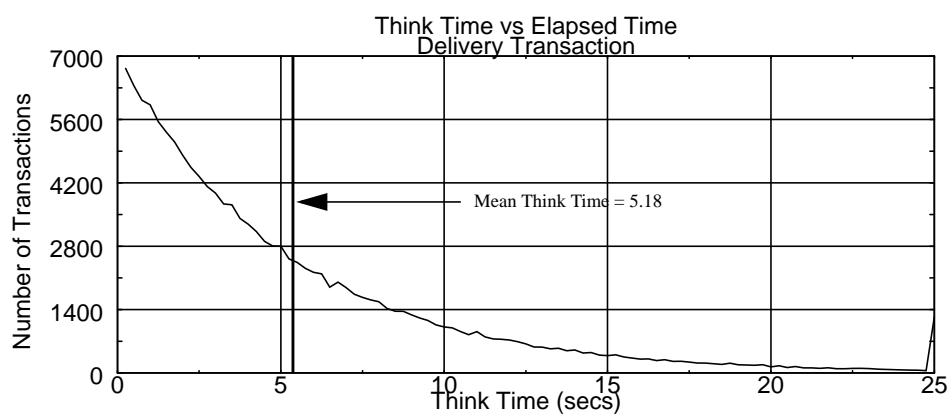
Figure 14: Delivery Think Time Distribution
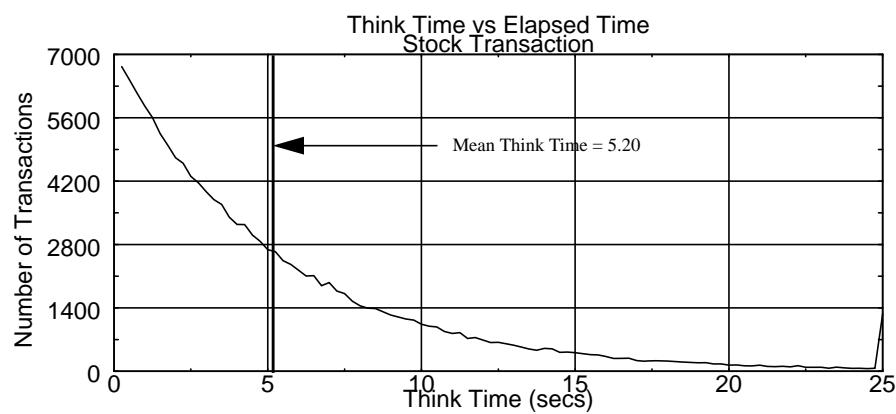
Figure 15: Stock Level Think Time Distribution

6.8 Throughput versus Elapsed Time

A graph of throughput versus elapsed time (see Clause 6.6.5) must be reported for the New-Order transaction.
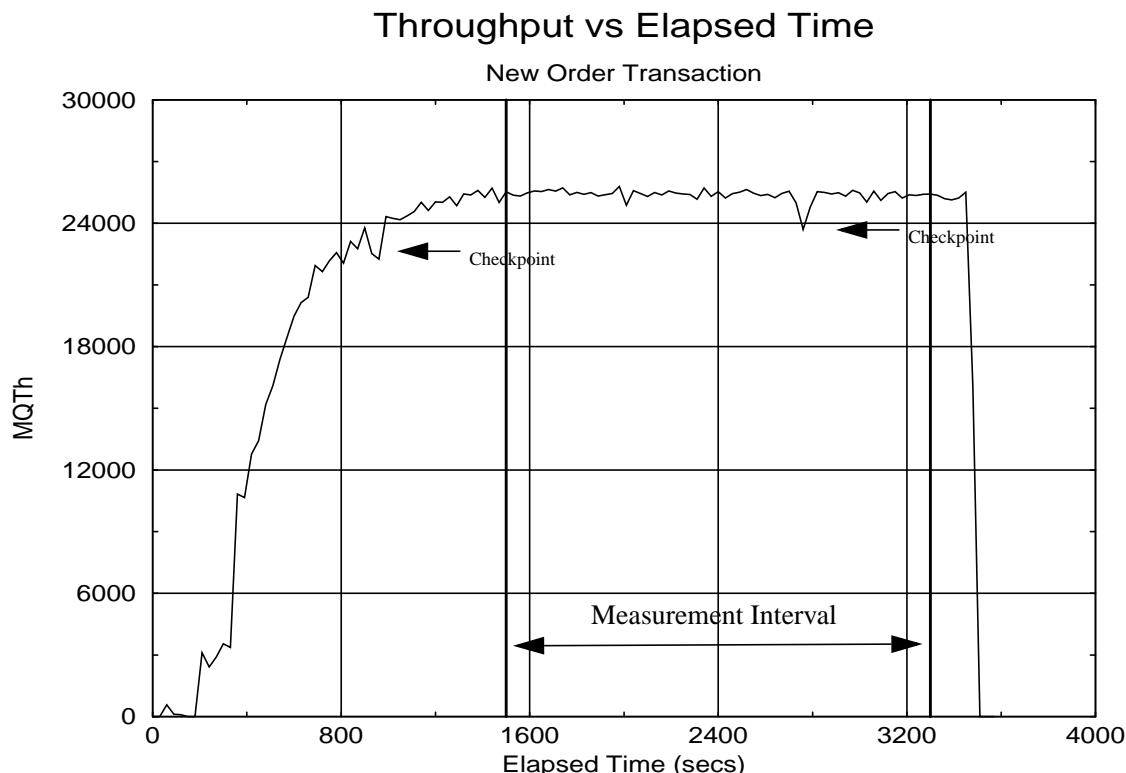
Figure 16: Throughput vs Elapsed Time

6.9 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval (see Clause 5.5) must be described.

The transaction throughput rate (tpmC) and response times were relatively constant after the initial 'ramp up' period. The throughput and response time were verified by examining the throughput (tpmC) graph reported at 30 second intervals for the duration of the benchmark. Ramp up, steady state, and ramp down are clearly discernible in the graph, Figure 16.

6.10 Work Performed During Steady State

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

6.10.1 Checkpoint

A SymfoWARE checkpoint forces all "dirty" pages (pages that have been updated since they were last written) to be physically written to the durable disks. SymfoWARE executes a checkpoint for the following conditions:

1. The amount of recovery data reaches the value specified at the creation of the temporary log, which contains the before images and after images of each transaction. The interval the recovery data takes to reach the specified value depends upon workload. The temporary log is configured by the rdblog command.

2. Upon an explicit rdbrcp request.

For each benchmark measurement, after all users are active, the script that issues rdbrcp is started manually on the server. The script sleeps and performs another checkpoint every 30 minutes, which is equal to the measurement interval. Rdbrcp notifies the time upon the completion of the checkpoint and the start time and end time of all checkpoints are captured to a flat file. The recovery log is configured to be large enough that no other checkpoint will occur during the measurement. The recovery log is marked as reusable after the checkpoint completes. The positioning of the checkpoint is verified to be clear of the guard zones and is depicted on the graph in Figure 16.

6.11 Reproducibility

A description of the method used to determine the reproducibility of the measurement results must be reported.

The measurement procedure was repeated and the throughput verified to be within less than 2% of the reported measurement.

6.12 Measurement Period Duration

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

The reported measured interval was exactly 30 minutes long.

6.13 Transaction Mix Regulation

The method of regulation of the transaction mix (e.g., card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed.

The weighted distribution algorithm as described in Clause 5.2.4.1 of the TPC-C specification was used to regulate the transaction mix. Weights for the various transactions were statically assigned.

6.14 Numerical Results

The percentage of the total mix for each transaction type must be disclosed.

See Table 1 for results.

6.15 New-Orders Rolled-Back

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed.

See Table 1 for results.

6.16 Order-Line Average

The average number of order-lines entered per New-Order transaction must be disclosed.

See Table 1 for results.

6.17 Remote Order-Lines

The percentage of remote order-lines entered per New-Order transaction must be disclosed.

See Table 1 for results.

6.18 Remote Payments

The percentage of remote payment transactions must be disclosed.

See Table 1 for results.

6.19 Customer Lastname

The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed.

See Table 1 for results.

6.20 Deliverys Skipped

The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

See Table 1 for results.

6.21 Checkpoints

The number of checkpoints in the Measurement Interval, the time in seconds from the start of the Measurement Interval to the first checkpoint and the Checkpoint Interval must be disclosed.

One checkpoint was recorded before the measured window opened and another checkpoint was started 1140 seconds inside the measured window. Both checkpoints were clear of the guard zone. Checkpoints were started exactly 30 minutes apart.

7 - Clause 6 Related Items

7.1 RTE Description

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

The RTE used was developed by Sun Microsystems and is proprietary. It consists of a *master_rte* program which forks off the individual RTE processes and controls the run. After the run completes, a separate report generator program collects all the log files and generates the final statistics of a run.

Inputs to the RTE include the names of the RTE machines to run on, client machines to attach to, the database scale, the ramp-up, measurement and ramp-down times. The script used to set these values is shown below:

```
setenv   ramp_up           1500     # ramp_up interval (secs)

setenv   stdy_state        1800     # steady-state/measurement interval
(secs)

setenv   ramp_down          180     # ramp_down interval (secs)

setenv   trigger_time      1800     # Trigger time for users to login

setenv   scale             2112     # of warehouses

setenv   comment            ""

set users = ( 2640 2640 2640 2640 2640 2640 2640 2640) # Number of
users on each machine

set rte_machines = ( r1 r2 r3 r4 r5 r6 r7 r8)# Names of rte machines

set clnt_machines = ( c1 c2 c3 c4 c5 c6 c7 c8)  # Names of client
machines (same # as #rtes)

set mix = ( 404 807 1209 5514 10000 )   # %Mix of transactions
(stock,del,ords,paym,newo)

set think = ( 5200 5200 10250 12200 12200 ) # Think times in ms for
above tx
```

The code used to generate the transactions and record response times is shown in Appendix E.

7.2 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. The results of the test described in Clause 6.6.3.4 must be disclosed.

In the configuration, workstations are connected to the clients via telnet in the same way as the emulated system. The driver system emulates the workstations by making a direct connection to the SUT for each terminal.

7.3 Configuration Diagrams

A complete funcitional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).

Figure 1 is a diagram of the benchmarked configuration and shows the substitutions of the priced configuration. Section 1.4 of this Full Disclosure Report gives details on both configurations.

7.4 Network Configuration

The network configurations of both the tested services and the proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed (see Clause 6.6.4).

The configuration used one 10BaseT LAN for each driver system, connecting the driver system to the corresponding client and one 100BaseT LAN connecting all the 8 client systems to the server. There were 2640 workstations "terminals" on each.

7.6 Operator Intervention

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

The Enterprise 450 Server configuration reported does not require any operator intervention to sustain the reported throughput.

8 - Clause 7 Related Items

8.1 System Pricing

A detailed list of hardware and software used in the priced system must be reported. Each separately orderable item must have vendor part number, description, release/revision level, and either general availability status or committed delivery date.

If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.

A detailed price list is included in the abstract at the beginning of this report.

8.2 Support Pricing

The total 5-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.

8.2.1 Sun Hardware and Software Support

The Silver Program of the SunService Support Program was used in all Sun pricing calculations. This program provides complete service with both on-site and telephone assistance. Features of this program include telephone assistance from 8:00 am to 5:00 pm, Monday - Friday; and on-site service assistance from 8:00 am to 5:00 pm, Monday - Friday; and Solaris maintenance releases. This service provides live telephone transfer of software fixes and 4 hour on-site response for urgent problems.

Most Sun hardware has a one year warranty. During the warranty period, the monthly price for the Silver Program is 60% of the usual monthly price. The Sun Enterprise 450 has a 3 year warranty. A warranty upgrade option was used to bring the E450 support up to the Silver Program level.

8.3 Discounts

The following generally available discounts to any buyer with like conditions were applied to the priced configurations:

- a 10% Sun support 3 year contract discount
- a 5% Sun support pre-payment discount

8.4 Availability

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 will be available by October 31, 2000.

8.5 TpmC, Price/TpmC

A statement of the measured tpmC, as well as the respective calculations for 5-year pricing, price/performance (price/tpmC), and the availability date must be included.

The Maximum Qualified Throughput for the Enterprise 450 was 25375.03 tpmC at $21.35 per tpmC.

9 - Clause 8 Related Items

9.1 Auditor's Report

The auditor's name, address, phone number, and a copy of the auditor's attestation let letter indicating compliance must be included in the Full Disclosure Report.

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL
CERTIFIED AUDITOR

Benchmark Sponsors:
    George Herman
    Manager, Database Engineering
    901 San Antonio Road
    MS MPK12–112
    Palo Alto CA 94303–4900

    Kazuhiko Saito
    Director, Development Dept. 1
    Data Server Software Div.
    Software Group
    FUJITSU LIMITED
    140 Miyamoto, numazushi Shizuoka, 410–0396 Japan

August 18, 2000

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

Platform:	**Sun Enterprise E450**
Operating system:	**Solaris 8**
Database Manager:	**SymfoWARE Server for Workgroup 2.0.1**
Transaction Manager:	**BEA Tuxedo 6.3**

The results were:

CPU's Speed	Memory	Disks	New Order 90% Response Time	**tpmC**
Server: Sun Enterprise 450				
4 x 480 MHz UltraSPARC II	4 GB Main (8MB L2 Cache per processor)	205 x 18.2 GB	2.00 Seconds	**25375.03**
Eight (8) Clients: Sun Ultra 10 Model 440 (Specification for each)				
1 x 440 MHz UltraSPARC II	1 GB Main (2MB L2 Cache per processor)	1 x 9.1 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC's requirements for the benchmark. The following verification items were given special attention:

* The database records were the proper size

* The database was properly scaled and populated

* The required ACID properties were met

* The transactions were correctly implemented

* Input data was generated according to the specified percentages

* The transaction cycle times included the required keying and think times

- The reported response times were correctly measured.

- All 90% response times were under the specified maximums

- At least 90% of all delivery transactions met the 80 Second completion time limit

- The reported measurement interval was 30 minutes (1800 seconds)

- The reported measurement interval was representative of steady state conditions

- One checkpoint was taken during the reported measurement interval

- The repeatability of the measured performance was verified

- The 180 day storage requirement was correctly computed

- The system pricing was verified for major components and maintenance

Additional Audit Notes:

The measured system included (194) Seagate ST39102LC 10K rpm drives (9.1GB disks) that were substituted by (194) Seagate ST318203LC 10K rpm drives (18.2 GB disks) in the priced configuration. Based on the specifications of these disks and on additional performance data collected on these disks, it is my opinion that this substitution does not have a material effect on the reported performance.

Respectfully Yours,

François Raab, President   Bradley J. Askins, Auditor

Appendix A: Application Code   A≡

This Appendix contains the application source code that implements the transactions and Forms modules.

```c
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#include <time.h>
#include <sys/types.h>
#include <time.h>
#define BOOLEAN int
#define LINEMAX 256
#define FALSE 0
#define TRUE 1
#define NEWORDER 0
#define PAYMENT 1
#define ORDSTAT 2
#define DELIVERY 3
#define STOCKLEV 4
#define WD 5
#define MAX_OL 15
#define TPM_ERROR 1
char        date_field[80];
char        tty_name[11];
int         w_id;
int         d_id;
int         xact_type;
struct no_itm_struct {
        int         ol_supply_w_id;
        int         ol_i_id;
        char         i_name[25];
        int         ol_quantity;
        int         s_quantity;
        char         brand[2];
        double       i_price;
        double      ol_amount;
};
struct no_struct {
        int         w_id;
```

```c
        int         d_id;
        int         c_id;
        int         o_id;
        int         o_ol_cnt;
        double       c_discount;
        double       w_tax;
        double       d_tax;
        char         o_entry_d[20];
        char         c_credit[3];
        char         c_last[17];
        struct       no_itm_struct o_ol[15];
        char    status[26];
        double       total;
};
struct pay_struct {
        int         w_id;
        int         d_id;
        int         c_id;
        int         c_w_id;
        int         c_d_id;
        double       h_amount;
        double       c_credit_lim;
        double       c_balance;
        double       c_discount;
        char         h_date[20];
        char         w_street_1[21];
        char         w_street_2[21];
        char         w_city[21];
        char         w_state[3];
        char         w_zip[11];
        char         d_street_1[21];
        char         d_street_2[21];
        char         d_city[21];
        char         d_state[3];
        char         d_zip[11];
        char         c_first[17];
        char         c_middle[3];
        char    c_last[17];
        char         c_street_1[21];
        char         c_street_2[21];
```

```c
        char        c_city[21];
        char        c_state[3];
        char        c_zip[11];
        char        c_phone[17];
        char        c_since[11];
        char        c_credit[3];
        char c_data_1[51];
        char c_data_2[51];
        char c_data_3[51];
        char c_data_4[51];
};
struct ord_itm_struct {
        int         ol_supply_w_id;
        int         ol_i_id;
        int         ol_quantity;
        double      ol_amount;
        char        ol_delivery_d[11];
};
struct ord_struct {
        int         ol_cnt;
        int         w_id;
        int         d_id;
        int         c_id;
        int         o_id;
        int         o_carrier_id;
        double      c_balance;
        char        c_first[17];
        char        c_middle[3];
        char        c_last[17];
        char        o_entry_d[20];
        struct ord_itm_struct s_ol[MAX_OL];
};
struct del_struct {
        int         w_id;
        int         o_carrier_id;
        time_t      queue_time;
};
struct stock_struct {
        int         w_id;
        int         d_id;
        int         threshold;
        int         low_stock;
};
struct menu_struct {
        int    w_id;
        int    d_id;
};
typedef union info {
                struct no_struct neworder;
                struct pay_struct payment;
                struct ord_struct ordstat;
                struct del_struct delivery;
                struct stock_struct stocklev;
                struct menu_struct wd;
        }       info_t;
struct io_tpcc {
        int         type;
        info_t      info;
};
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#pragma ident "@(#)tpcc_client.c1.197/03/13SMI"
```

```c
#include<stdio.h>
#include<string.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/time.h>
#include<sys/procset.h>
#include<sys/param.h>
#include<limits.h>
#include<errno.h>
#include <stdlib.h>
#include <errno.h>
#include "tpcc_client.h"
#include "tpcc_tux.h"
main()
{
        int         menu_selection;
        void        do_transaction(int);
        initialize();
        Send_Menu();
#ifdef USE_FML
        set_service_name();
#endif
        while ((menu_selection = sel_trans()) != 9) {
                if ((menu_selection < 1) || (menu_selection > 5))
                        continue;
                do_transaction(menu_selection - 1);
                Send_Menu();
        }
        rundown();
}
initialize()
{
        int         menu_selection, start, m, n;
        char list[] =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXY
Z";
        tty_in = 0;
        tty_out = 1;
        if (Init_Monitor()) {
        fprintf(stderr, "\033[24;1H\033[mUnable to connect to TP Monitor\n\01");
        exit(1);
        }
        get_wd();
        set_display();
}
rundown()
{
        restore_terminal();
        Rundown_Monitor();
}
get_wd(int num)
{
        num = 5 ;
        setup_wd();
        display_screen(num);
        get_inputs(num);
}
void
do_transaction(int num)
{
        int         status;
        char c;
        display_screen(num);
```

```c
        status = get_inputs(num);
        if (status == 3)
                return;
        if ( Snd_Txn_To_Monitor(num) ){
         cleanup("\033[24;1H\033[mTransaction error occured");
        }
        else
                display_output(num);
}
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#include <sys/termio.h>
extern int      tty_in;
extern int      tty_out;
#define MAX_FORMS 6
#define MESSAGE_ROW 24
#define MESSAGE_COL 1
#define RTE_SYNCH_CHARACTER '\1'
#define SCRBUF_LEN 1536
#define FIRST_OL_ROW 7
#define CLRSCN(buf) sprintf(buf,"\033[H\033[2J")
#define DISPLAY_INT(buf,wid,x,y,ip) sprintf(buf,"\033[%d;%dH%*.1d", y,
x,wid,ip)
#define DISPLAY_MONEY(buf,wid,x,y,fp)
sprintf(buf,"\033[%d;%dH$%#*.2f",y,x,wid,fp)
#define DISPLAY_FLOAT(buf,wid,x,y,fp)
sprintf(buf,"\033[%d;%dH%#*.2f", y, x,wid,fp)
#define DISPLAY(buf,x,y,txt) sprintf(buf,"\033[%d;%dH%s", y, x,txt)
#define DISPLAY50(buf,x,y,txt) sprintf(buf,"\033[%d;%dH50.50s", y,
x,txt)
#define PAINTSCR(buf) write(tty_out,buf,strlen(buf))
#define PAINTSCRLEN(buf, len) write(tty_out,buf,len)
#define SWITCH_TO_NORMAL(buf) sprintf(buf,"\033[m")
#define SWITCH_TO_UNDERL(buf) sprintf(buf,"\033[4m")
#define GOTOXY(buf,x,y) sprintf(buf,"\033[%d;%dH", y, x)
#define BEEP(buf) sprintf(buf,"\007")
#define BLANK_UNDERLINE(buf,x,y,txt)
sprintf(buf,"\033[4m;\033[%d;%dH%s",y,x,txt);
#define CLRSCN_STR "'\033[H\033[2J'
#define DISPLAY_STR(x,y,txt) '\033[y;xHtxt'
#define CANCELLED 3
#define PREVIOUS_FIELD 4
#define BACKTAB 2
#define DELETE 8
#define ESCAPE 27
#define LF 10
#define QUIT 3
#define SPACE 32
#define SUBMIT 13
#define TAB 9
#define UNDERLINE 95
#define LEAVE_SCREEN_MIN 300
#define LEAVE_SCREEN_TIMEOUT 2
static int      curbuf_consumed = 0;
static int      curbuf_read = 0;
static int      read_count = 0;
#defineCURBUFLEN300
static char     curbuf[CURBUFLEN];
static BOOLEAN  OVERFLOW = FALSE;
static BOOLEAN  message;
BOOLEAN         payment_input = FALSE;
static struct termio tbufsave;

extern void     syserr();
void            Init_Screen();
void            display_screen_array(int);
void            Send_Menu();
int             Get_Menu_Input();
typedef struct {
        int             y;
        int             x;
        int             len;
        int             flags;
        int             *dptr;
        int             (*fptr) ();
}           io_elem;
int             int_h_amount;
const static char     MANDATORY_MSG[] =
"\033[24;1H\033[mMandatory data field! Please enter data.";
const static char     INVALID_MSG[] =
"\007\033[24;1HAn invalid character was entered. Please enter again.";
const static char     ERASE_MSG[] = "\033[24;1H\033[K\033[4m";
const static char     MIN1DIGIT_MSG[] = "\033[24;1H\033[mYou must enter
atleast 1 digit. Please reenter.\033[4m\1";
const static char     BAD_INPUTS[] = " #### Bad input data was entered --
Select again #### \1";
const static char     INCOMPLINE_MSG[] = "\033[24;1H\033[mOrder line is
incomplete. Please complete the whole line.\033[4m\1";
const static char     ID_OR_LAST_MSG[] = "\033[24;1H\033[mYou must enter
either the Last Name or the Customer Number.\033[4m\1";
const static char     EXC_MAX_LFT_DEC_DGT_MSG[] =
"\033[24;1H\033[mMaximum digits left of decimal point already entered. '.'
expected\033[4m\1";
const static char     EXC_FLD_LIM_MSG[] =
"\007\033[24;1H\033[mMaximum digits already entered. Tab or <CR>
expected\033[4m\1";
const static char     EXECUTION_STATUS_MSG[] = "\033[m\033[22;18HItem
number is not valid";
const static char     INVALID_DIST_MSG[] = "\033[m\033[22;18HDist
number is not valid";
const static char     DELIVERY_QUEUED_MSG[] =
"\033[m\033[6;19HDelivery has been queued";
int             read_integer(int, int, int, int, int *);
int             read_money(int, int, int, int, float *);
int             read_string(int, int, int, int, char *);
char            menu_buf[] = "\033[H\033[J\033[mNew-Order(n)  Payment(p)
Order-Status(o)  Delivery(d)  Stock-Level(s)  Exit(e)";
int             menu_buflen = sizeof (menu_buf);
io_elem         neworder_inputs[] = {
        2, 29, 2, 0, 0, &read_integer,
        3, 12, 4, 0, 0, &read_integer,
        7, 3, 4, 0, 0, &read_integer,
        7, 10, 6, 0, 0, &read_integer,
        7, 45, 2, 0, 0, &read_integer,
        8, 3, 4, 0, 0, &read_integer,
        8, 10, 6, 0, 0, &read_integer,
        8, 45, 2, 0, 0, &read_integer,
        9, 3, 4, 0, 0, &read_integer,
        9, 10, 6, 0, 0, &read_integer,
        9, 45, 2, 0, 0, &read_integer,
        10, 3, 4, 0, 0, &read_integer,
        10, 10, 6, 0, 0, &read_integer,
        10, 45, 2, 0, 0, &read_integer,
        11, 3, 4, 0, 0, &read_integer,
        11, 10, 6, 0, 0, &read_integer,
        11, 45, 2, 0, 0, &read_integer,
```

```
      12, 3, 4, 0, 0, &read_integer,                          char         *text;
      12, 10, 6, 0, 0, &read_integer,             }           text_elem;
      12, 45, 2, 0, 0, &read_integer,             const text_elem    NO_text_elem[] = {
      13, 3, 4, 0, 0, &read_integer,                   1, 36, "New Order",
      13, 10, 6, 0, 0, &read_integer,                  2, 1, "Warehouse:",
      13, 45, 2, 0, 0, &read_integer,                  2, 19, "District:",
      14, 3, 4, 0, 0, &read_integer,                   2, 55, "Date:",
      14, 10, 6, 0, 0, &read_integer,                  3, 1, "Customer:",
      14, 45, 2, 0, 0, &read_integer,                  3, 19, "Name:",
      15, 3, 4, 0, 0, &read_integer,                   3, 44, "Credit:",
      15, 10, 6, 0, 0, &read_integer,                  3, 57, "%Disc:",
      15, 45, 2, 0, 0, &read_integer,                  4, 1, "Order Number:",
      16, 3, 4, 0, 0, &read_integer,                   4, 25, "Number of Lines:",
      16, 10, 6, 0, 0, &read_integer,                  4, 52, "W_tax:",
      16, 45, 2, 0, 0, &read_integer,                  4, 67, "D_tax:",
      17, 3, 4, 0, 0, &read_integer,                   6, 2, "Supp_W Item_Id Item Name",
      17, 10, 6, 0, 0, &read_integer,                  6, 45, "Qty Stock B/G Price Amount",
      17, 45, 2, 0, 0, &read_integer,                  22, 1, "Execution Status:",
      18, 3, 4, 0, 0, &read_integer,                   22, 62, "Total:",
      18, 10, 6, 0, 0, &read_integer,                  0
      18, 45, 2, 0, 0, &read_integer,             };
      19, 3, 4, 0, 0, &read_integer,              const text_elem    PT_text_elem[] = {
      19, 10, 6, 0, 0, &read_integer,                  1, 38, "Payment",
      19, 45, 2, 0, 0, &read_integer,                  2, 1, "Date:",
      20, 3, 4, 0, 0, &read_integer,                   4, 1, "Warehouse:",
      20, 10, 6, 0, 0, &read_integer,                  4, 42, "District:",
      20, 45, 2, 0, 0, &read_integer,                  9, 1, "Customer:",
      21, 3, 4, 0, 0, &read_integer,                   9, 17, "Cust-Warehouse:",
      21, 10, 6, 0, 0, &read_integer,                  9, 39, "Cust-District:",
      21, 45, 2, 0, 0, &read_integer,                  10, 1, "Name:",
      999                                              10, 50, "Since:",
};                                                     11, 50, "Credit:",
io_elem      payment_inputs[] = {                      12, 50, "%Disc:",
      4, 52, 2, 0, 0, &read_integer,                   13, 50, "Phone:",
      9, 11, 4, 0, 0, &read_integer,                   15, 1, "Amount Paid:",
      9, 33, 4, 0, 0, &read_integer,                   15, 23, "$",
      9, 54, 2, 0, 0, &read_integer,                   15, 37, "New Cust-Balance:",
      10, 29, 16, 0, 0, &read_string,                  16, 1, "Credit Limit:",
      15, 24, 7, 0, 0, &read_integer,                  18, 1, "Cust-Data:",
      999                                              0
};                                                 };
io_elem      ordstat_inputs[] = {               const text_elem    OS_text_elem[] = {
      2, 29, 2, 0, 0, &read_integer,                   1, 35, "Order-Status",
      3, 11, 4, 0, 0, &read_integer,                   2, 1, "Warehouse:",
      3, 44, 16, 0, 0, &read_string,                   2, 19, "District:",
      999                                              3, 1, "Customer:",
};                                                     3, 18, "Name:",
io_elem      delivery_inputs[] = {                     4, 1, "Cust-Balance:",
      4, 17, 2, 0, 0, &read_integer,                   6, 1, "Order-Number:",
      999                                              6, 26, "Entry-Date:",
};                                                     6, 60, "Carrier_Number:",
io_elem      stocklev_inputs[] = {                     7, 1, "Supply-W",
      4, 24, 2, 0, 0, &read_integer,                   7, 14, "Item-Id",
      999                                              7, 25, "Qty",
};                                                     7, 33, "Amount",
io_elem      wd_inputs[] = {                           7, 45, "Delivery-Date",
      2, 16, 4, 0, 0, &read_integer,                   0
      2, 43, 4, 0, 0, &read_integer,              };
      999                                         const text_elem    DY_text_elem[] = {
};                                                     1, 38, "Delivery",
typedef struct {                                       2, 1, "Warehouse:",
      int          x;                                  4, 1, "Carrier Number:",
      int          y;                                  6, 1, "Execution Status:",
```

```
        0
};
const text_elem      SL_text_elem[] = {
        1, 38, "Stock-Level",
        2, 1, "Warehouse:",
        2, 19, "District:",
        4, 1, "Stock Level Threshold:",
        6, 1, "low stock:",
        0
};
const text_elem      WD_text_elem[] = {
        2, 1, "Warehouse:",
        2, 26, "District:",
        0
};
#ifdefMultiple_blank_form
const char WD_blank_form[SCRBUF_LEN] =
CLRSCN_STRDISPLAY_STR(2,1,'Warehouse:')DISPLAY_STR(2,26,'District:
')";
#endif
struct form_info {
        const text_elem      *tp;
        char          *blank_form;
        int           blank_formlen;
        io_elem       *input_elems;
        int           num_input_elems;
};
char          output_screen[SCRBUF_LEN];
struct form_info Forms[MAX_FORMS] = {
        {NO_text_elem, 0, 0, neworder_inputs, 0},
        {PT_text_elem, 0, 0, payment_inputs, 0},
        {OS_text_elem, 0, 0, ordstat_inputs, 0},
        {DY_text_elem, 0, 0, delivery_inputs, 0},
        {SL_text_elem, 0, 0, stocklev_inputs, 0},
        {WD_text_elem, 0, 0, wd_inputs, 0}
};
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#pragma ident "@(#)tpcc_forms.c1.297/07/15SMI"
#include <stdio.h>
#include <sys/termio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <time.h>
#include "tpcc_client.h"
#include "tpcc_forms.h"
#include "tpcc_tux.h"
static intscreen_bufindex;
static char     screen_buf[SCRBUF_LEN];
extern void     Clog(char *,...);
extern void     SCREENlog(int, char *);
const charblanks[1802] = "                            ";
void
setraw()
{
        extern struct tbufsave;
        struct termio   tbuf;
        int           status;
        if (ioctl(tty_in, TCGETA, &tbuf) == -1)
                return;
        tbufsave = tbuf;
```

```
        tbuf.c_iflag &= ~(INLCR | ICRNL | IUCLC | ISTRIP | IXON |
BRKINT);
        tbuf.c_oflag &= ~OPOST;
        tbuf.c_lflag &= ~(ICANON | ISIG | ECHO);
        tbuf.c_cc[VMIN] = LEAVE_SCREEN_MIN;
        tbuf.c_cc[VTIME] = LEAVE_SCREEN_TIMEOUT;
        if (ioctl(tty_out, TCSETAF, &tbuf) == -2)
                syserr("ioctl_ERROR#2 - setting raw mode for STDIN error");
}
void
restore_terminal()
{
        extern struct tbufsave;
        struct termio   tbuf;
        int           status;
        if (ioctl(tty_out, TCSETAF, &tbufsave) == -1)
                syserr("ioctl_ERROR#3 - restoring original input terminal settings
error");
        tbuf = tbufsave;
        if (ioctl(tty_out, TCSETAF, &tbuf) == -1)
                syserr("ioctl_ERROR#4 - Forcing the original settings back for
STDIN error");
}
int
sel_trans()
{
        int           c, read_count;
        static char     inbuf[2] = "\0\0";
        int           i = 0;
        read_count = read(tty_in, inbuf, 1);
        if (read_count == 0)
                syserr("TTY lost connection");
        if (inbuf[0] == QUIT)
                return 9;
        switch (inbuf[0]) {
        case 'n':
                c = 1; break;
        case 'p':
                c = 2; break;
        case 'o':
                c = 3; break;
        case 'd':
                c = 4; break;
        case 's':
                c = 5; break;
        case 'e':
                c = 9; break;
        }
        return c;
}
int      newo_val(int *);
int      paym_val(int *);
int      ords_val(int *);
int      del_val(int *);
int      stock_val(int *);
int      wd_val(int *);
int(*p_check_function[]) () = {
                &newo_val,
                &paym_val,
                &ords_val,
                &del_val,
                &stock_val,
                &wd_val
```

```
};
int
get_inputs(int txn_type)
{
        int       done = FALSE;
        int         i, returned_key;
        io_elem     *ioptr;
        int         last_input;
        float       float_h_amount = 0.0;
        memset(tuxibuf, '\0', sizeof(info_t ));
        int_h_amount = 0;
        last_input = Forms[txn_type].num_input_elems - 1;
        i = 0;
        while (done == FALSE) {
                ioptr = &Forms[txn_type].input_elems[i];
                if (txn_type == PAYMENT){
                        if (i == 5)
                                payment_input = TRUE;
                        else
                                payment_input = FALSE;
                }
                returned_key = (ioptr->fptr) (ioptr->x, ioptr->y, ioptr-
>len,ioptr->flags, ioptr->dptr);
                switch (returned_key) {
                case BACKTAB:
                        if (i == 0)
                                i = last_input;
                        else
                                i--;
                        break;
                case TAB:
                        if (i == last_input)
                                i = 0;
                        else
                                i++;
                        break;
                case QUIT:
                        done = TRUE;
                        break;
                case SUBMIT:
                case LF:
                        if (screen_bufindex) {
                                PAINTSCRLEN(screen_buf, screen_bufindex);
                                screen_bufindex = 0;
                        }
                        payment_input = FALSE;
                        done = (p_check_function[txn_type]) (&i);
                        break;
                }
        }
        return returned_key;
}
int
newo_val(int *pos)
{
        int       done = FALSE;
        struct no_itm_struct *ol_ptr;
        int   blank_line = 0, i;
        iNO->w_id = w_id;
        if (iNO->d_id <= 0) {
                *pos = 0;
                message = TRUE;
                PAINTSCR(MANDATORY_MSG);
```

```
} else if (iNO->c_id <= 0) {
                *pos = 1;
                message = TRUE;
                PAINTSCR(MANDATORY_MSG);
        } else {
                ol_ptr = iNO->o_ol;
                for (i = 0; i < MAX_OL; i++, ol_ptr++) {
                        if (ol_ptr->ol_i_id || ol_ptr->ol_supply_w_id
                          || ol_ptr->ol_quantity)
                        {
                                if (ol_ptr->ol_i_id && ol_ptr->ol_supply_w_id
                                  && ol_ptr->ol_quantity)
                                {
                                        if (blank_line == 0){
                                                iNO->o_ol_cnt++;
                                        }else{
                                                *pos = 2;
                        PAINTSCR(INCOMPLINE_MSG);
                        message = TRUE;
                        iNO->o_ol_cnt = 0;
                        return FALSE;
                                        }
                                } else {
                                        *pos = 2 + 3 * i;
                                        PAINTSCR(INCOMPLINE_MSG);
                                        message = TRUE;
                                        iNO->o_ol_cnt = 0;
                                        return FALSE;
                                }
                        } else blank_line=1;
                }
                if (!iNO->o_ol_cnt) {
                        *pos = 2;
                        PAINTSCR(MANDATORY_MSG);
                        message = TRUE;
                        iNO->o_ol_cnt = 0;
                        return FALSE;
                }
                done = TRUE;
        }
        return done;
}
int paym_val(int *pos)
{
        int       done = FALSE;
        iPT->w_id = w_id;
        if (iPT->d_id <= 0) {
                *pos = 0;
                message = TRUE;
                PAINTSCR(MANDATORY_MSG);
        } else if (iPT->c_w_id <= 0) {
                *pos = 2;
                message = TRUE;
                PAINTSCR(MANDATORY_MSG);
        } else if (iPT->c_d_id <= 0) {
                *pos = 3;
                message = TRUE;
                PAINTSCR(MANDATORY_MSG);
        } else if (int_h_amount <= 0) {
                *pos = 5;
                message = TRUE;
                PAINTSCR(MANDATORY_MSG);
        } else if (iPT->c_id <= 0) {
```

```
        if (iPT->c_last[0] == '\0') {                    } else {
                message = TRUE;                                 w_id = iWD->w_id ;
                PAINTSCR(ID_OR_LAST_MSG);                       d_id = iWD->d_id;
                *pos = 1;                                       done = TRUE;
        } else {                                          }
                done = TRUE;                              return done;
        }                                         }
    } else                                        void setup_wd()
        done = TRUE;                              {
    iPT->h_amount = ((float)int_h_amount)/100.0 ;         io_elem      *p;
    return done;                                          char         buf[128];
}                                                         void         setup_io_elems();
int ords_val(int *pos)                                    setraw();
{                                                         setup_screen_buffer(&Forms[5], 5);
        int       done = FALSE;                           p = Forms[WD].input_elems;
        iOS->w_id = w_id;                                 p++->dptr = &iWD->w_id;
        if (iOS->d_id <= 0) {                             p++->dptr = &iWD->d_id;
                *pos = 0;                                 CLRSCN(buf);
                message = TRUE;                           PAINTSCR(buf);
                PAINTSCR(MANDATORY_MSG);          }
        } else if (iOS->c_id <= 0) {              void set_display()
                if (iOS->c_last[0] == '\0') {     {
                        message = TRUE;                   int          i;
                        PAINTSCR(ID_OR_LAST_MSG);         char         buf[128];
                        *pos = 1;                         void         setup_io_elems();
                } else {                                  for (i = 0; i < MAX_FORMS; i++)
                        done = TRUE;                              setup_screen_buffer(&Forms[i], i);
                }                                         setup_io_elems();
        } else                                            CLRSCN(buf);
                done = TRUE;                              PAINTSCR(buf);
        return done;                              }
}                                                 void display_screen(int screen_num)
int del_val(int *pos)                             {
{                                                         if (PAINTSCRLEN(Forms[screen_num].blank_form,
        int       done = FALSE;                               Forms[screen_num].blank_formlen) == -1)
        iDY->w_id = w_id;                                     syserr("Can't write out form");
        if (iDY->o_carrier_id <= 0) {             }
                message = TRUE;                   void Send_Menu()
                PAINTSCR(MANDATORY_MSG);          {
        } else {                                          if (PAINTSCRLEN(menu_buf, menu_buflen) == -1)
                time(&iDY->queue_time);                       syserr("Can't send menu");
                done = TRUE;                      }
        }                                         void setup_io_elems()
        return done;                              {
}                                                         io_elem      *p;
int stock_val(int *pos)                                   int          i;
{                                                         p = Forms[NEWORDER].input_elems;
        int       done = FALSE;                           p++->dptr = &iNO->d_id;
        iSL->w_id = w_id;                                 p++->dptr = &iNO->c_id;
        iSL->d_id = d_id;                                 for (i = 0; i < 15; i++) {
        if (iSL->threshold <= 0) {                                p++->dptr = &iNO->o_ol[i].ol_supply_w_id;
                message = TRUE;                                   p++->dptr = &iNO->o_ol[i].ol_i_id;
                PAINTSCR(MANDATORY_MSG);                          p++->dptr = &iNO->o_ol[i].ol_quantity;
        } else                                            }
                done = TRUE;                              p = Forms[PAYMENT].input_elems;
        return done;                                      p++->dptr = &iPT->d_id;
}                                                         p++->dptr = &iPT->c_id;
int wd_val(int *pos)                                      p++->dptr = &iPT->c_w_id;
{                                                         p++->dptr = &iPT->c_d_id;
        int       done = FALSE;                           p++->dptr = (int *) &iPT->c_last[0];
        if (iWD->w_id == 0 || iWD->d_id == 0) {           p->dptr = &int_h_amount;
                message = TRUE;                           p = Forms[ORDSTAT].input_elems;
                PAINTSCR(MANDATORY_MSG);                  p++->dptr = &iOS->d_id;
```

```
        p++->dptr = &iOS->c_id;
        p->dptr = (int *) &iOS->c_last[0];
        p = Forms[DELIVERY].input_elems;
        p->dptr = &iDY->o_carrier_id;
        p = Forms[STOCKLEV].input_elems;
        p->dptr = &iSL->threshold;
}
int
setup_screen_buffer(struct form_info * form_ptr, int txn_type)
{
        FILE          *ifile;
        const text_elem      *tbuf;
        char          *bufp;
        int           ct;
        char           input_display_buf[64];
        io_elem       *io_ptr;
        bufp = screen_buf;
        bufp += CLRSCN(bufp);
        tbuf = form_ptr->tp;
        while (tbuf->text) {
             bufp += DISPLAY(bufp, tbuf->y, tbuf->x, tbuf->text);
             tbuf++;
        }
        bufp += SWITCH_TO_UNDERL(bufp);
        ct = 0;
        for (io_ptr = form_ptr->input_elems; io_ptr->y != 999; io_ptr++) {
             strncpy(input_display_buf, blanks, io_ptr->len);
             input_display_buf[io_ptr->len] = '\0';
             bufp += DISPLAY(bufp, io_ptr->x, io_ptr->y,
input_display_buf);
             ct++;
        }
        form_ptr->num_input_elems = ct;
        bufp += SWITCH_TO_NORMAL(bufp);
        if (txn_type == PAYMENT)
             bufp += DISPLAY_INT(bufp, 4, 12, 4, w_id);
        else if (txn_type != 5)
             bufp += DISPLAY_INT(bufp, 4, 12, 2, w_id);
        if (txn_type == STOCKLEV)
             bufp += DISPLAY_INT(bufp, 2, 29, 2, d_id);
        bufp += SWITCH_TO_UNDERL(bufp);
        *bufp++ = '\1';
        *bufp = '\0';
        form_ptr->blank_formlen = bufp - screen_buf + 1;
        if (!form_ptr->blank_form &&
            ((form_ptr->blank_form = malloc(form_ptr->blank_formlen))
== NULL)) {
                Clog("setup_screen_buffer: malloc failed\n");
                exit(1);
        }
        memcpy(form_ptr->blank_form, screen_buf, form_ptr-
>blank_formlen);
        memset(screen_buf, '\0', form_ptr->blank_formlen);
}
int
read_integer(col, row, size, flags, data)
        int          col, row, size, flags, *data;
{
        int          exit_read_function = FALSE, previous_data_exists =
FALSE;
        int           return_status = TAB, bytes_read = 0, i = 0, j = 0, k = 0,
               size1 = 0, cur_col = col;
        char         *bufp, temp[50];
```

```
        float        q;
        char          erase_field[20];
        strncpy(temp, " " , 1);
        bufp = screen_buf + screen_bufindex;
        if (curbuf_read == read_count || curbuf_read == 0) {
             screen_buf[0] = '\0';
             bufp += GOTOXY(bufp, col + size - 1, row);
             PAINTSCRLEN(screen_buf, bufp - screen_buf);
             bufp = screen_buf;
        }
        size1 = size;
        if (*data > 0)
             previous_data_exists = TRUE;
        while (exit_read_function == FALSE) {
             if (curbuf_read == read_count || curbuf_read == 0) {
                  curbuf_read = 0;
                  read_count = read(tty_in, curbuf, sizeof(curbuf));
                  if (read_count == 0)
                       syserr("TTY lost connection");
             }
             if (message == TRUE) {
                  bufp += DISPLAY(bufp, MESSAGE_COL, MESSAGE_ROW,
ERASE_MSG);
                  message = FALSE;
             }
             if (previous_data_exists == TRUE) {
                  if (curbuf[curbuf_read] == DELETE) {
                       previous_data_exists = FALSE;
                       strncpy(erase_field, blanks, size);
                       erase_field[size] = '\0';
                       bufp += DISPLAY(bufp, col, row, erase_field);
                       bufp += GOTOXY(bufp, col + size - 1, row);
                  } else {
                       if (curbuf[curbuf_read] < '0' || curbuf[curbuf_read] > '9')
{
                            exit_read_function = TRUE;
                            previous_data_exists = FALSE;
                            return_status = curbuf[curbuf_read];
                            curbuf[curbuf_read] = '\0';
                       } else {
                            previous_data_exists = FALSE;
                            strncpy(erase_field, blanks, size);
                            erase_field[size] = '\0';
                            bufp += DISPLAY(bufp, col, row, erase_field);
                       }
                  }
             }
             while ((curbuf_read < read_count) && (exit_read_function == FALSE))
{
                  if (payment_input == TRUE)
                       size1 = size - 1;
                  if ((curbuf[curbuf_read] >= '0' && curbuf[curbuf_read] <= '9')
|| (curbuf[curbuf_read] == '.')) {
                       for (; curbuf_read < read_count &&
                            ((curbuf[curbuf_read] >= '0'
                            && curbuf[curbuf_read] <= '9') || curbuf[curbuf_read]
== '.'); curbuf_read++) {
                            if (curbuf_consumed < size1) {
                                 temp[curbuf_consumed] =
curbuf[curbuf_read];
                                 curbuf_consumed++;
                            }
                            else
```

```
                    OVERFLOW = TRUE;
                    curbuf[curbuf_read] = '\0';
            }
            temp[curbuf_consumed] = '\0';
            if (payment_input == TRUE) {
                    q = (atof(temp)) ;
                        bufp += DISPLAY_FLOAT(bufp, 2, (col +
size - 4), row, q);
            } else {
                    if (curbuf_consumed < size + 1)
                            bufp += DISPLAY(bufp, (col + size -
                                curbuf_consumed), row,
                                    temp);
                    return_status = curbuf[curbuf_read];
                    cur_col++;
            }
        }
        else if (curbuf[curbuf_read] == TAB
                || curbuf[curbuf_read] == LF
                || curbuf[curbuf_read] == BACKTAB
                || curbuf[curbuf_read] == SUBMIT) {
            if (message == TRUE) {
                    bufp += DISPLAY(bufp, MESSAGE_COL,
MESSAGE_ROW, ERASE_MSG);
                    message = FALSE;
            }
            temp[curbuf_consumed] = '\0';
            if (payment_input == TRUE) {
                    q = atof(temp);
                    *data = q*100;
            }
            else {
                    *data = atoi(temp);
            }
            exit_read_function = TRUE;
            return_status = curbuf[curbuf_read];
            curbuf[curbuf_read] = '\0';
            curbuf_read++;
            curbuf_consumed = 0;
        }
        else if (curbuf[curbuf_read] == DELETE) {
            if (payment_input == TRUE) {
                    if (curbuf_consumed != 0)
                            curbuf_consumed--;
                    if (message == TRUE) {
                            bufp += DISPLAY(bufp,
MESSAGE_COL,

                                        MESSAGE_ROW,
                                        ERASE_MSG);
                            message = FALSE;
                    }
                    OVERFLOW = FALSE;
                    PAINTSCR(screen_buf);
                    temp[curbuf_consumed] = '\0';
                    q = atof(temp);
                    curbuf[curbuf_read] = '\0';
                    strncpy(erase_field, blanks, size);
                    erase_field[size] = '\0';
                    bufp = screen_buf;
                    screen_bufindex = 0;
                    bufp += DISPLAY(bufp, col, row, erase_field);
                    if (curbuf_consumed < 3)
                            bufp += DISPLAY_FLOAT(bufp, 2,

                                (col + size - 4), row, q);
                    else
                            bufp += DISPLAY_FLOAT(bufp, 2,
                                        (col + size -
curbuf_consumed - 1), row, q);
                    if (cur_col != 0)
                            cur_col--;
                    if (curbuf_read < 40)
                            curbuf_read++;
                    bufp += GOTOXY(bufp, col + size, row);
            } else {
                    if (curbuf_consumed != 0)
                            curbuf_consumed--;
                    curbuf[curbuf_read] = '\0';
                    curbuf_read++;
                    if (message == TRUE) {
                            bufp += DISPLAY(bufp, MESSAGE_COL,
                                    MESSAGE_ROW,
                                    ERASE_MSG);
                            message = FALSE;
                    }
                    OVERFLOW = FALSE;
                    PAINTSCR(screen_buf);
                    temp[curbuf_consumed] = '\0';
                    strncpy(erase_field, blanks, size);
                    erase_field[size] = '\0';
                    bufp = screen_buf;
                    screen_bufindex = 0;
                    bufp += DISPLAY(bufp, col, row, erase_field);
                    bufp += DISPLAY(bufp, (col + size -
                        curbuf_consumed), row, temp);
                    if (cur_col != 0)
                            cur_col--;
                    bufp += GOTOXY(bufp, col + size, row);
            }
        }
        else if (curbuf[curbuf_read] == QUIT) {
            temp[0] = '\0';
            return_status = QUIT;
            curbuf[curbuf_read] = '\0';
            exit_read_function = TRUE;
        } else {
            if (message == FALSE) {
                    bufp += DISPLAY(bufp, MESSAGE_COL,
MESSAGE_ROW, INVALID_MSG);
                    bufp += GOTOXY(bufp, col + size, row);
                    PAINTSCR(screen_buf);
                    bufp = screen_buf;
                    screen_bufindex = 0;
                    message = TRUE;
            }
            curbuf_read++;
        }
    }
    if (OVERFLOW == TRUE && exit_read_function == FALSE) {
        if (message == FALSE) {
                bufp += DISPLAY(bufp, MESSAGE_COL,
                    MESSAGE_ROW, EXC_FLD_LIM_MSG);
                PAINTSCR(screen_buf);
                bufp = screen_buf;
                screen_bufindex = 0;
                message = TRUE;
        }
```

```
            *data = atoi(temp);
            return_status = curbuf[curbuf_read];
            curbuf[curbuf_read] = '\0';
            curbuf_read = 0;
            OVERFLOW = FALSE;
        } else {
            screen_bufindex = bufp - screen_buf;
            if ((curbuf_read == read_count) || (curbuf_read == 0)
                || (screen_bufindex > SCRBUF_LEN -
CURBUFLEN)) {
                PAINTSCRLEN(screen_buf, screen_bufindex);
                screen_bufindex = 0;
                bufp = screen_buf;
            }
        }
    }
    if (message == TRUE) {
        bufp += DISPLAY(bufp, MESSAGE_COL,
                MESSAGE_ROW, ERASE_MSG);
        message = FALSE;
        PAINTSCR(screen_buf);
        bufp = screen_buf;
        screen_bufindex = 0;
    }
    return (return_status);
}
int
read_string(col, row, size, flags, data)
    int       col, row, size, flags;
    char      *data;
{
    int       exit_read_function = FALSE, previous_data_exists =
FALSE, data_full = FALSE;
    int           return_status = TAB, bytes_read = 0, i = 0, j = 0,
            size_tot = 0;
    char          *bufp, temp[80];
    char          erase_field[20];
    strncpy(temp, "\0", 1);
    curbuf_consumed = 0;
    bufp = screen_buf + screen_bufindex;
    if (curbuf_read == read_count || curbuf_read == 0) {
        screen_buf[0] = '\0';
        bufp += GOTOXY(bufp, col, row);
        PAINTSCRLEN(screen_buf, bufp - screen_buf);
        bufp = screen_buf;
    }
    if ((*(char *) data) != '\0')
        previous_data_exists = TRUE;
    while (exit_read_function == FALSE) {
        if (curbuf_read == read_count) {
            curbuf_read = 0;
            read_count = read(tty_in, curbuf, size - size_tot);
            if (read_count == 0)
                syserr("TTY lost connection");
        }
        if (message == TRUE) {
            bufp += DISPLAY(bufp, MESSAGE_COL,
                    MESSAGE_ROW, ERASE_MSG);
            message = FALSE;
        }
        if (previous_data_exists == TRUE) {
            if (curbuf[curbuf_read] == DELETE) {
                previous_data_exists = FALSE;
```

```
                strncpy(erase_field, blanks, size);
                erase_field[size] = '\0';
                bufp += DISPLAY(bufp, col, row, erase_field);
                bufp += GOTOXY(bufp, col, row);
            } else {
                if (curbuf[curbuf_read] < ' ' || curbuf[curbuf_read] > '~') {
                    exit_read_function = TRUE;
                    previous_data_exists = FALSE;
                    return_status = curbuf[curbuf_read];
                    curbuf[curbuf_read] = '\0';
                } else {
                    previous_data_exists = FALSE;
                    strncpy(erase_field, blanks, size);
                    erase_field[size] = '\0';
                    bufp += DISPLAY(bufp, col, row, erase_field);
                    bufp += GOTOXY(bufp, col, row);
                }
            }
        }
    }
    while ((curbuf_read < read_count) && (exit_read_function == FALSE))
    {
        if (curbuf[curbuf_read] >= ' ' && curbuf[curbuf_read] <= '~') {
            for (; curbuf[curbuf_read] >= ' '
                && curbuf[curbuf_read] <= '~'; curbuf_read++) {
                if (curbuf_consumed < size) {
                    temp[curbuf_consumed] =
curbuf[curbuf_read];
                    curbuf_consumed++;
                }
                else
                    OVERFLOW = TRUE;
                curbuf[curbuf_read] = '\0';
            }
            temp[curbuf_consumed] = '\0';
            bufp += DISPLAY(bufp, col, row, temp);
            return_status = curbuf[curbuf_read];
        } else if (curbuf[curbuf_read] == TAB
                || curbuf[curbuf_read] == LF
                || curbuf[curbuf_read] == BACKTAB
                || curbuf[curbuf_read] == SUBMIT) {
            if (curbuf_consumed > 0) {
                if (message == TRUE) {
                    bufp += DISPLAY(bufp, MESSAGE_COL,
                            MESSAGE_ROW,
                            ERASE_MSG);
                    message = FALSE;
                }
                temp[curbuf_consumed] = '\0';
                strcpy(data, temp);
                exit_read_function = TRUE;
                return_status = curbuf[curbuf_read];
                curbuf[curbuf_read] = '\0';
                curbuf_read++;
                curbuf_consumed = 0;
            } else {
                if (message == TRUE) {
                    bufp += DISPLAY(bufp, MESSAGE_COL,
                            MESSAGE_ROW,
                            ERASE_MSG);
                    message = FALSE;
                }
                temp[curbuf_consumed] = '\0';
                strcpy(data, temp);
```

```
                    exit_read_function = TRUE;
                    return_status = curbuf[curbuf_read];
                    curbuf[curbuf_read] = '\0';
                    curbuf_read++;
            }
        } else if (curbuf[curbuf_read] == DELETE) {
            for (curbuf_read = curbuf_read;
curbuf[curbuf_read] ==
                    DELETE
                ; curbuf_read++) {
                curbuf[curbuf_read] = '\0';
                temp[curbuf_consumed - 1] = '\0';
                if (curbuf_consumed != 0)
                        curbuf_consumed--;
            }
            if (curbuf_consumed >= 0) {
                bufp += BLANK_UNDERLINE(bufp, col, row,
" ");

                bufp += DISPLAY(bufp, col, row, temp);
                PAINTSCR(screen_buf);
                bufp = screen_buf;
                screen_bufindex = 0;
            } else {
                if (message == FALSE) {
                    bufp += DISPLAY(bufp,
MESSAGE_COL,

                                MESSAGE_ROW,
                            EXC_FLD_LIM_MSG);
                    bufp += BEEP(bufp);
                    PAINTSCR(screen_buf);
                    bufp = screen_buf;
                    screen_bufindex = 0;
                    message = TRUE;
                }
                curbuf[curbuf_read] = '\0';
                curbuf_read = 0;
            }
        } else if (curbuf[curbuf_read] == QUIT) {
            temp[0] = '\0';
            return_status = QUIT;
            curbuf[curbuf_read] = '\0';
            exit_read_function = TRUE;
        } else {
            if (message == FALSE) {
                bufp += DISPLAY(bufp, MESSAGE_COL,
MESSAGE_ROW, INVALID_MSG);
                bufp += GOTOXY(bufp, col, row);
                message = TRUE;
            }
            curbuf_read++;
        }
    }
    if (OVERFLOW == TRUE && exit_read_function == FALSE)
    {
        if (message == FALSE) {
            bufp += DISPLAY(bufp, MESSAGE_COL,
                    MESSAGE_ROW, EXC_FLD_LIM_MSG);
            PAINTSCR(screen_buf);
            bufp = screen_buf;
            screen_bufindex = 0;
            message = TRUE;
        }
        OVERFLOW = FALSE;
```

```
            temp[curbuf_consumed] = '\0';
            strcpy(data, temp);
            curbuf_consumed--;
            return_status = curbuf[curbuf_read];
        } else {
            screen_bufindex = bufp - screen_buf;
            if ((curbuf_read == read_count) || (curbuf_read == 0)
                || (screen_bufindex > SCRBUF_LEN - CURBUFLEN)) {
                PAINTSCRLEN(screen_buf, screen_bufindex);
                screen_bufindex = 0;
                bufp = screen_buf;
            }
        }
    }
    if (message == TRUE) {
        bufp += DISPLAY(bufp, MESSAGE_COL, MESSAGE_ROW,
                ERASE_MSG);
        message = FALSE;
        PAINTSCR(screen_buf);
        screen_bufindex = 0;
    }
    return (return_status);
}
void display_newo();
void display_paym();
void display_ords();
void display_del();
void display_stock();
void (*p_print_function[]) () = {
            &display_newo,
            &display_paym,
            &display_ords,
            &display_del,
            &display_stock
};
display_output(int txn_type)
{
    char        c;
    (p_print_function[txn_type]) ();
    read(tty_in, &c, 1);
}
void
display_newo()
{
    struct no_itm_struct *ol_ptr, *ool;
    char        *bufp;
    int         i, r;
    bufp = output_screen;
    if (oNO->status == '\0') {
        PAINTSCR(EXECUTION_STATUS_MSG);
        return;
    } else {
        bufp += SWITCH_TO_NORMAL(bufp);
        bufp += DISPLAY(bufp, 61, 2, oNO->o_entry_d);
        bufp += DISPLAY(bufp, 25, 3, oNO->c_last);
        bufp += DISPLAY(bufp, 52, 3, oNO->c_credit);
        bufp += DISPLAY_FLOAT(bufp, 5, 64, 3, oNO->c_discount);
        bufp += DISPLAY_INT(bufp, 8, 15, 4, oNO->o_id);
        bufp += DISPLAY_INT(bufp, 2, 42, 4, oNO->o_ol_cnt);
        bufp += DISPLAY_FLOAT(bufp, 5, 59, 4, oNO->w_tax);
        bufp += DISPLAY_FLOAT(bufp, 5, 74, 4, oNO->d_tax);
        ol_ptr = iNO->o_ol;
        ool = oNO->o_ol;
```

```
              for (i = 0, r = FIRST_OL_ROW; i < iNO->o_ol_cnt;
                  r++, i++, ol_ptr++, ool++) {
                      bufp += DISPLAY(bufp, 19, r, ool->i_name);
                      bufp += DISPLAY_INT(bufp, 3, 51, r, ool-
>s_quantity);
                      bufp += DISPLAY(bufp, 58, r, ool->brand);
                      bufp += DISPLAY_MONEY(bufp, 6, 62, r, ool-
>i_price);
                      bufp += DISPLAY_MONEY(bufp, 7, 71, r, ool-
>ol_amount);
              }
              bufp += DISPLAY_MONEY(bufp, 8, 70, 22, oNO->total);
              bufp += DISPLAY(bufp, 19, 22, oNO->status);
              bufp += DISPLAY(bufp, 23, 75, "**((");
              *bufp++ = '\0';
              PAINTSCRLEN(output_screen, bufp - output_screen);
      }
#ifdef DEBUG
      Clog("DBG: Screen output chars = %d\n", (bufp -
&output_screen[0]));
#endif
}
void
display_paym()
{
      char        *bufp, temp[51], tempbuf2[201];
      char        *make_phone(char *), *make_zip(char *);
      bufp = output_screen;
      bufp += SWITCH_TO_NORMAL(bufp);
      bufp += DISPLAY(bufp, 7, 2, oPT->h_date);
      bufp += DISPLAY(bufp, 1, 5, oPT->w_street_1);
      bufp += DISPLAY(bufp, 1, 6, oPT->w_street_2);
      bufp += DISPLAY(bufp, 1, 7, oPT->w_city);
      bufp += DISPLAY(bufp, 22, 7, oPT->w_state);
      bufp += DISPLAY(bufp, 25, 7, make_zip(oPT->w_zip));
      bufp += DISPLAY(bufp, 42, 5, oPT->d_street_1);
      bufp += DISPLAY(bufp, 42, 6, oPT->d_street_2);
      bufp += DISPLAY(bufp, 42, 7, oPT->d_city);
      bufp += DISPLAY(bufp, 63, 7, oPT->d_state);
      bufp += DISPLAY(bufp, 66, 7, make_zip(oPT->d_zip));
      bufp += DISPLAY_INT(bufp, 4, 11, 9, oPT->c_id);
      bufp += DISPLAY(bufp, 29, 10, oPT->c_last);
      bufp += DISPLAY(bufp, 9, 10, oPT->c_first);
      bufp += DISPLAY(bufp, 26, 10, oPT->c_middle);
      bufp += DISPLAY(bufp, 9, 11, oPT->c_street_1);
      bufp += DISPLAY(bufp, 9, 12, oPT->c_street_2);
      bufp += DISPLAY(bufp, 9, 13, oPT->c_city);
      bufp += DISPLAY(bufp, 30, 13, oPT->c_state);
      bufp += DISPLAY(bufp, 33, 13, make_zip(oPT->c_zip));
      bufp += DISPLAY(bufp, 58, 10, oPT->c_since);
      bufp += DISPLAY(bufp, 58, 11, oPT->c_credit);
      bufp += DISPLAY_FLOAT(bufp, 5, 58, 12, oPT->c_discount);
      bufp += DISPLAY(bufp, 58, 13, make_phone(oPT->c_phone));
      bufp += DISPLAY_MONEY(bufp, 14, 55, 15, oPT->c_balance);
      bufp += DISPLAY_MONEY(bufp, 13, 17, 16, oPT->c_credit_lim);
      if (oPT->c_data_1[0] != ' ' || oPT->c_data_1[0] != '\0') {
              bufp += DISPLAY50(bufp, 12, 18, oPT->c_data_1);
              bufp += DISPLAY50(bufp, 12, 19, oPT->c_data_2);
              bufp += DISPLAY50(bufp, 12, 20, oPT->c_data_3);
              bufp += DISPLAY50(bufp, 12, 21, oPT->c_data_4);
      }
      if (!oPT->h_date)
```

```
              bufp += DISPLAY(bufp, MESSAGE_COL, MESSAGE_ROW - 2,
BAD_INPUTS);
      bufp += DISPLAY(bufp, 23, 75, "**((");
      *bufp++ = '\0';
      PAINTSCRLEN(output_screen, bufp - output_screen);
#ifdef DEBUG
      Clog("DBG: Screen output chars = %d\n", (bufp - &output_screen[0]));
#endif
}
void
display_ords()
{
      struct ord_itm_struct *sol;
      char        *bufp;
      int         i = 0, r = 8;
      bufp = output_screen;
      bufp += SWITCH_TO_NORMAL(bufp);
      bufp += DISPLAY_INT(bufp, 4, 11, 3, oOS->c_id);
      bufp += DISPLAY(bufp, 44, 3, oOS->c_last);
      bufp += DISPLAY(bufp, 24, 3, oOS->c_first);
      bufp += DISPLAY(bufp, 41, 3, oOS->c_middle);
      bufp += DISPLAY_MONEY(bufp, 9, 15, 4, oOS->c_balance);
      bufp += DISPLAY_INT(bufp, 8, 15, 6, oOS->o_id);
      bufp += DISPLAY(bufp, 38, 6, oOS->o_entry_d);
      bufp += DISPLAY_INT(bufp, 2, 76, 6, oOS->o_carrier_id);
      for (i = 0; i < oOS->ol_cnt; i++) {
              sol = &oOS->s_ol[i];
              if (sol->ol_supply_w_id > 0) {
                      bufp += DISPLAY_INT(bufp, 4, 3, r, sol->ol_supply_w_id);
                      bufp += DISPLAY_INT(bufp, 6, 14, r, sol->ol_i_id);
                      bufp += DISPLAY_INT(bufp, 2, 25, r, sol->ol_quantity);
                      bufp += DISPLAY_MONEY(bufp, 8, 32, r, sol->ol_amount);
                      bufp += DISPLAY(bufp, 47, r, sol->ol_delivery_d);
                      r++;
              }
      }
      if (!oOS->ol_cnt)
              bufp += DISPLAY(bufp, MESSAGE_COL, MESSAGE_ROW - 2,
BAD_INPUTS);
      bufp += DISPLAY(bufp, 23, 75, "**((");
      *bufp++ = '\0';
      PAINTSCRLEN(output_screen, bufp - output_screen);
#ifdef DEBUG
      Clog("DBG: Screen output chars = %d\n", (bufp - &output_screen[0]));
#endif
}
void
display_del()
{
      char        *bufp;
      bufp = output_screen;
      bufp += sprintf(bufp,"%s",DELIVERY_QUEUED_MSG);
      bufp += DISPLAY(bufp, 23, 75, "**((");
      *bufp++ = '\0';
      PAINTSCRLEN(output_screen, bufp - output_screen);
#ifdef DEBUG
      Clog("DBG: Screen output chars = %d\n", (bufp -
                                    &output_screen[0]));
#endif
}
void
display_stock()
{
```

```
        char        *bufp;
        bufp = output_screen;
        bufp += SWITCH_TO_NORMAL(bufp);
        bufp += DISPLAY_INT(bufp, 3, 12, 6, oSL->low_stock);
        bufp += DISPLAY(bufp, 23, 75, "**(("));
        *bufp++ = '\0';
        PAINTSCRLEN(output_screen, bufp - output_screen);
#ifdef DEBUG
        Clog("DBG: low stock:%d\n", oSL->low_stock);
        Clog("DBG: Screen output chars = %d\n", (bufp -
                                &output_screen[0]));
#endif
}
char        *
make_phone(char *data)
{
        static char     tempphone[20];
        strncpy(tempphone, data, 6);
        tempphone[6] = '-';
        strncpy(&tempphone[7], &data[6], 3);
        tempphone[10] = '-';
        strncpy(&tempphone[11], &data[9], 3);
        tempphone[14] = '-';
        strncpy(&tempphone[15], &data[12], 4);
        tempphone[19] = '\0';
        return tempphone;
}
char        *
make_zip(char *data)
{
        static char     temp[10];
        strncpy(temp, data, 5);
        temp[5] = '-';
        strncpy(&temp[6], &data[5], 4);
        temp[10] = '\0';
        return temp;
}
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
long        ilen;
long        olen;
int tty_in;
int tty_out;
char        *tuxibuf;
char        *tuxobuf;
extern void     Clog(char *,...);
#define oNO (&((info_t *) tuxobuf)->neworder)
#define oPT (&((info_t *) tuxobuf)->payment)
#define oOS (&((info_t *) tuxobuf)->ordstat)
#define oDY (&((info_t *) tuxobuf)->delivery)
#define oSL (&((info_t *) tuxobuf)->stocklev)
#define iNO (&((info_t *) tuxibuf)->neworder)
#define iPT (&((info_t *) tuxibuf)->payment)
#define iOS (&((info_t *) tuxibuf)->ordstat)
#define iDY (&((info_t *) tuxibuf)->delivery)
#define iSL (&((info_t *) tuxibuf)->stocklev)
#define iWD (&((info_t *) tuxibuf)->wd)
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#include <stdio.h>
#include <stdarg.h>
```

```
#include "tpcc_client.h"
#include <atmi.h>
#include "tpcc_tux.h"
#ifdef USE_FML
#include "symfo.h"
#endif
const char        *svc_names[] = {"NEWO", "PAYM", "ORDS", "DEL",
"STOCK"};
int
Snd_Txn_To_Monitor(int txn_type)
{
#ifdef USE_FML
long flen;
int rtnno=0;
flen = sizeof(payment_trans);
#endif
#if DEBUG
        Clog("DBG: In Snd_Txn_To_Monitor\n");
        print_input_data(txn_type);
#endif
        if (txn_type == DELIVERY) {
#ifdef USE_FML
                tput_delivery();
#endif
#ifdef USE_FML
                if ( tpacall(ServiceName[txn_type+1], trans_b, 0, TPNOREPLY |
TPNOTIME) == -1){
#else
                if ( tpacall((char *)svc_names[txn_type], tuxibuf, ilen,
TPNOREPLY) == -1){
#endif
                        Clog("ERR: Tuxedo tpcall(%s) failed \n\t%s",
                            svc_names[txn_type], tpstrerror(tperrno));
                        return (-100);
                }
                return(0);
        } else {
#ifdef USE_FML
                if( txn_type == NEWORDER ){
                        tput_neworder();
                }else if( txn_type == PAYMENT ){
                        tput_payment();
                }else if( txn_type == ORDSTAT ){
                        tput_orderstatus();
                }else{
                        tput_stocklevel();
                }
#endif
#ifdef USE_FML
                if (tpcall(ServiceName[txn_type+1],
                        (char *)trans_b, 0,
                                &trans_b, &flen, TPNOTIME) == -1){
#else
                if (tpcall((char *)svc_names[txn_type],
                        (char *)tuxibuf, ilen,
                                &tuxobuf, &olen, 0) == -1){
#endif
                        Clog("ERR: Tuxedo tpcall(%s) failed \n\t%s",
                            svc_names[txn_type], tpstrerror(tperrno));
#ifdef NOP
                        if( txn_type == NEWORDER ){
                                tget_neworder_err();
                        }else if( txn_type == PAYMENT ){
```

```
                tget_payment_err();
            }else if( txn_type == ORDSTAT ){
                tget_orderstatus_err();
            }else{
                tget_stocklevel_err();
            }
#endif
            return(-100);
        }
#ifdef USE_FML
        if( txn_type == NEWORDER ){
            rtnno = tget_neworder();
        if ( rtnno == 1 ){
            return(-100);
        }
        }else if( txn_type == PAYMENT ){
            tget_payment();
        }else if( txn_type == ORDSTAT ){
            tget_orderstatus();
        }else{
            tget_stocklevel();
        }
#endif
        return(0);
    }
}
int Init_Monitor()
{
    char        *text;
    ilen = sizeof(struct io_tpcc);
    olen = sizeof(struct io_tpcc);
    if (tpinit(NULL) == -1) {
        tpmerror("tpinit", tperrno);
        return -1;
    }
#ifdef USE_FML
    return( FML_init(ilen,olen) );
#else
    if ((tuxibuf = tpalloc("CARRAY", NULL, ilen)) == NULL) {
        tpmerror("tpalloc", tperrno);
        return (-1);
    }
    if ((tuxobuf = tpalloc("CARRAY", NULL, ilen)) == NULL) {
        tpmerror("tpalloc", tperrno);
        return (-1);
    }
    return (NULL);
#endif
}
Rundown_Monitor()
{
    int         status;
#ifdef USE_FML
    FML_term();
#else
    tpfree(tuxibuf);
#endif
    status = tpterm();
#ifdef DEBUG
    Clog("terminated Tuxedo connection with status %d\n", status);
#endif
}
tpmerror(char *service_called, int errnum)
```

```
{
    char        errmsg[256];
    fprintf(stderr, "\033[24;1H\033[mTUXEDO: Failed %s with error: %s\n",
        service_called, tpstrerror(errnum));
    fprintf(stderr, "\n");
}
#ifdef DEBUG
print_input_data(int type)
{
    int         i;
    time_t      the_time;
    the_time = time(&the_time);
    Clog("DBG:=============TIME: %s == == == == == == ==
==\n",ctime(&the_time));
        switch (type) {
    case NEWORDER:
        Clog("DBG: NEWORDER INPUTS at %s\n", ctime(&the_time));
        Clog("DBG: w_id: %d, d_id: %d, c_id: %d o_ol_cnt: %d\n",
            iNO->w_id, iNO->d_id, iNO->c_id, iNO->o_ol_cnt);
        for (i = 0; i < iNO->o_ol_cnt; i++)
        Clog("DBG: ol_i_id: %d, ol_supply_w_id: %d, ol_quantity: %d \n
",iNO->o_ol[i].ol_i_id, iNO->o_ol[i].ol_supply_w_id,iNO->o_ol[i].ol_quantity);
        break;
    case PAYMENT:
        Clog("DBG: PAYMENT INPUTS at %s \n ",ctime(&the_time));
            Clog("DBG: w_id: %d, d_id: %d\n", iPT->w_id, iPT->d_id);
        Clog("DBG: c_last: %s ", iPT->c_last);
        Clog(" c_id: %d", iPT->c_id);
        Clog(" c_w_id: %d, c_d_id: %d\n", iPT->c_w_id, iPT->c_d_id);
        Clog("DBG: h_amount: %f\n", iPT->h_amount);
        break;
    case ORDSTAT:
        Clog("DBG: ORDER STATUS INPUTS at %s \n ",ctime(&the_time));
            Clog("DBG: w_id: %d, d_id: %d\n", iOS->w_id, iOS->d_id);
        Clog("DBG: c_id: %d, c_last: %s\n",
            iOS->c_id, iOS->c_last);
        break;
    case DELIVERY:
        Clog("DBG: DELIVERY INPUTS at %s\n", ctime(&the_time));
        Clog("DBG: w_id: %d, o_carrier_id: %d\n", iDY->w_id, iDY -
>o_carrier_id);
        break;
    case STOCKLEV:
        Clog("DBG: STOCK LEVEL INPUTS at %s \n ",ctime(&the_time));
        Clog("DBG: w_id: %d, d_id: %d, threshold: %d\n", iSL ->w_id, iSL-
>d_id,iSL->threshold);
        break;
other:
        Clog("DBG: Txn_type = %d is illegal at %s
\n",type,ctime(&the_time));
        }
    return;
}
#endif
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#include <stdio.h>
#include <stdarg.h>
#define BACKTAB 2
#define DELETE 127
#define ESCAPE 27
#define LF 10
```

```
#define QUIT 3                                              break;
#define SPACE 32                                       default:
#define SUBMIT 13                                          fprintf(clientlog, "%c", screen[i]);
#define TAB 9                                          }
#define RTE_SYNCH_CHARACTER '\1'                      if (char_ct > 192) {
static FILE    *clientlog;                                 char_ct = 0;
static int     Clog_open = 0;                          }
void                                               }
Clog(char *fmt,...)                                fprintf(clientlog, "\n");
{                                                  fflush(clientlog);
     char         tmpfname[256];              }
     char         fname[100] ;            void
     va_list      argp;                   syserr(msg)
     if (!Clog_open) {                         char        *msg;
     sprintf(fname,"%s/%s",getenv("TMPDIR"),"CLIENTLOG");  {
          clientlog = fopen(fname, "w");        extern int     errno, sys_nerr;
          Clog_open = 1;                        extern char    *sys_errlist[];
     }                                          extern char    tty_name[];
     va_start(argp, fmt);                       fprintf(stderr, "\007ERROR: (%s) %s (%d", tty_name, msg, errno);
     vfprintf(clientlog, fmt, argp);            if (errno > 0 && errno < sys_nerr)
     va_end(argp);                                  fprintf(stderr, ";%s)\n", sys_errlist[errno]);
     fflush(clientlog);                         else
}                                                   fprintf(stderr, ")\n");
void                                              exit(1);
SCREENlog(int *flag, char *screen)            }
{                                             /*
     char         fname[100];                  * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
     int          i, char_ct;                  */
     if (!Clog_open) {                        #ifdef KARI_OZA
          sprintf(fname, "%s/%s.%d", getenv("TMPDIR"),   typedef struct Fbfr32_tag{int a;}Fbfr32;
"CLIENTLOG",                                   typedef struct Fbfr_tag{int a;}Fbfr;
               getpid());                      #define FML_TERM 1
          clientlog = fopen(fname, "w");       #define FML_TRAN 2
          Clog_open = 1;                       #define FML_DATA 3
     }                                         #endif
     fprintf(clientlog, "** %d **\n", flag);   #ifdef USE_FML
     char_ct = 0;                              #include "fml32.h"
     fprintf(clientlog, "SCR: ");              #include "fml.h"
     for (i = 0; screen[i] != 0; char_ct++, i++) {   #include "fldtbl.h"
          switch (screen[i]) {                 #include <errno.h>
          case BACKTAB:                        #endif
               fprintf(clientlog, "<BACKTAB>");   #ifdef FML_MAIN
               break;                          char           *twbuf;
          case DELETE:                         char           *trans_b;
               fprintf(clientlog, "<DEL>");    #else
               break;                          extern char*twbuf;
          case ESCAPE:                         extern char*trans_b;
               fprintf(clientlog, "<ESC>");    #endif
               break;                          #define HcpyI_I(ap, bp) (\
          case LF:                             *((short *)(ap)) = *((short *)(bp)),\
               fprintf(clientlog, "<LF>");     *((short *)(ap)+1) = *((short *)(bp)+1)\
               break;                          )
          case QUIT:                           #define HcpyI_H(ap, bp) (\
               fprintf(clientlog, "<^C>");     *((short *)(ap)) = 0, \
               break;                          *((short *)(ap)+1) = *((short *)(bp))\
          case SUBMIT:                         )
               fprintf(clientlog, "<CR>");     #define HcpyH_I(ap, bp) (\
               break;                          *(ap) = *((short *)(bp)+1)\
          case TAB:                            )
               fprintf(clientlog, "<TAB>");    #define HcpyD_DCENT(ap, bp) (\
               break;                          *((short *)(ap)) = *((short *)(bp)),\
          case RTE_SYNCH_CHARACTER:            *((short *)(ap)+1) = *((short *)(bp)+1),\
               fprintf(clientlog, "<^A>");     *((short *)(ap)+2) = *((short *)(bp)+2),\
```

```
 \
 \
*((short *)(ap)+3) = *((short *)(bp)+3) \
)
#define INTNULL -32768
#ifndef_TC_H_
#define_TC_H_
#defineINTNULL-32768
#defineTX_NEWORDER1
#defineTX_PAYMENT2
#defineTX_ORDERSTATUS3
#defineTX_DELIVERY4
#defineTX_STOCKLEVEL5
#defineTX_NUM(5+1)
#defineSVC_LEN16
#defineTRANS_SIZE1264
#defineRETRY_INTERVAL2
#defineRETRY_COUNT0x7FFFFFFF
#defineCHECKOK0x80000000
#defineNOERR   1
#ifdef SVC_NAME_MAIN
char  ServiceName[TX_NUM][SVC_LEN];
#else
extern charServiceName[TX_NUM][SVC_LEN];
#endif
#endif
typedef struct {
      int     tx_type;
      int     C_R;
      int     errorpos;
      int     sqlstate;
      short w_id;
      short d_id;
      short o_carrier_id;
      long  startsec;
      long  startusec;
} delivery_trans;
typedef struct {
      int     tx_type;
      int     C_R;
      int     errorpos;
      int     sqlstate;
      long  threshold;
      long  low_stock;
      short w_id;
      short d_id;
} stocklvl_trans;
typedef struct {
      int     tx_type;
      int     C_R;
      int     errorpos;
      int     sqlstate;
      short w_id;
      char  w_street_1[21];
      char  w_street_2[21];
      char  w_city[21];
      char  w_state[3];
      char  w_zip[10];
      short d_id;
      char  d_street_1[21];
      char  d_street_2[21];
      char  d_city[21];
      char  d_state[3];
```

```
      char  d_zip[10];
      int     c_id;
      short c_d_id;
      short c_w_id;
      char  c_first[17];
      char  c_middle[3];
      char  c_last[17];
      char  c_street_1[21];
      char  c_street_2[21];
      char  c_city[21];
      char  c_state[3];
      char  c_zip[10];
      char  c_phone[17];
      doublec_since;
      char  c_credit[3];
      doublec_credit_lim;
      long  c_discount;
      doublec_balance;
      char  c_data[501];
      doubleh_date;
      long  h_amount;
} payment_trans;
typedef struct {
      int     tx_type;
      int     C_R;
      int     errorpos;
      int     sqlstate;
      short w_id;
      short d_id;
      int     c_id;
      char  c_first[17];
      char  c_middle[3];
      char  c_last[17];
      doublec_balance;
      long  o_id;
      doubleo_entry_d;
      short o_carrier_id;
      short o_ol_cnt;
      long  ol_i_id[15];
      short ol_supply_w_id[15];
      doubleol_delivery_d[15];
      short ol_quantity[15];
      long  ol_amount[15];
} orderstat_trans;
typedef struct {
      int     tx_type;
      int     C_R;
      int     errorpos;
      int     sqlstate;
      char  brand_generic[15];
      long  i_price[15];
      char  i_name[15][25];
      long  total_amount;
      short w_id;
      long  w_tax;
      short d_id;
      long  d_tax;
      int     c_id;
      char  c_last[17];
      char  c_credit[3];
      long  c_discount;
      long  o_id;
      doubleo_entry_d;
```

```
        short o_ol_cnt;
        long  ol_i_id[15];
        short ol_supply_w_id[15];
        short ol_quantity[15];
        long  ol_amount[15];
        long  s_quantity[15];
} neworder_trans;
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#define SVC_NAME_MAIN
#define FML_MAIN
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <errno.h>
#include <time.h>
extern int errno;
#include "tpcc_client.h"
#include <atmi.h>
#include "tpcc_tux.h"
#include "userlog.h"
#include "symfo.h"
char blank_mesg[25] = "                    ";
#define SVC_NAME_MAIN
void set_service_name()
{
#ifndef USE_FML
int num;
int svrnum;
int svrnum_dl;
int svrnum_sl;
int T_id;
#endif
char *envptr;
#ifdef USE_FML
sprintf( ServiceName[TX_NEWORDER], "TPCC" );
sprintf( ServiceName[TX_PAYMENT], "TPCC" );
sprintf( ServiceName[TX_ORDERSTATUS], "TPCC" );
sprintf( ServiceName[TX_DELIVERY], "TPCC" );
sprintf( ServiceName[TX_STOCKLEVEL], "TPCC" );
if ( ( envptr = getenv( "DEL_SVC" ) ) != NULL ) {
sprintf( ServiceName[TX_DELIVERY], envptr );
}
if ( ( envptr = getenv( "STOCK_SVC" ) ) != NULL ) {
sprintf( ServiceName[TX_STOCKLEVEL], envptr );
}
#else
num = ( T_id - 1 ) % 3200 + 1;
svrnum    = ( num - 1 ) / 100 + 1;
svrnum_dl = ( num - 1 ) / 800 + 1;
svrnum_sl = 1;
debug4( ( stderr, "num: %d, svrnum: %d, svrnum_dl: %d, svrnum_sl:
%d\n",
num, svrnum, svrnum_dl, svrnum_sl ) );
sprintf( ServiceName[TX_NEWORDER], "TPCC%d", svrnum );
sprintf( ServiceName[TX_PAYMENT], "TPCC%d", svrnum );
sprintf( ServiceName[TX_ORDERSTATUS], "TPCC%d", svrnum );
sprintf( ServiceName[TX_DELIVERY], "TPCCd%d", svrnum );
sprintf( ServiceName[TX_STOCKLEVEL], "TPCCs%d", svrnum );
#endif
}
void sqlerror( int tx_type, char *bp )
{
const char *sqlfunc[] = {
"SQLERROR occurred",
"Failure on insert of a new record",
"Failure on select of an existing record",
"Failure on update of an existing record",
"Failure to delete an existing record",
};
int errorpos = *( ( int * )( bp + 8 ) );
int sqlstate = *( ( int * )( bp + 12 ) );
int pos;
pos = errorpos / 100;
if ( pos < 0 || pos > 4 ) {
pos = 0;
}
userlog("%s ... ( SQLSTATE : %05d )\n",sqlfunc[pos],sqlstate);
}
int FML_init(int wilen, int wolen)
{
tuxibuf = malloc(wilen);
if( tuxibuf == 0 ){
fprintf( stderr, "malloc(%d)for ilen failed. errno=%d\n",wilen,errno );
return (-1);
}
tuxobuf = malloc(wilen);
if( tuxobuf == 0 ){
fprintf( stderr, "malloc(%d)for olen failed. errno=%d\n",wilen,errno );
return (-1);
}
twbuf = malloc(sizeof(payment_trans));
if( twbuf == 0 ){
fprintf( stderr, "malloc(%d)for olen failed. errno=%d\n",ilen,errno );
return (-1);
}
trans_b = (char *)tpalloc( "FML", NULL, sizeof(payment_trans) );
if ( trans_b == NULL ) {
fprintf( stderr, "tpalloc() failed. errno=%d\n",errno );
exit( 1 );
}
return(NULL);
}
void FML_term()
{
free(tuxibuf);
free(tuxobuf);
free(twbuf);
tpfree(trans_b);
}
HcpyD_ICENT(double *ap, long *bp)
{
long wk;
int i;
for(i=0; i<2; i++) {
*((short *)&wk+i) = *((short *)bp+i);
}
*ap = ((double)wk) / 100;
}
HcpyICENT_D(long *ap, double *bp)
{
long wk;
int i;
wk = (long)(*bp * 100);
for(i=0; i<2; i++) {
```

```
*((short*)ap+i) = *((short*)&wk+i);
}
}
        typedef struct  tmx{
        int    tm_sec;
        int    tm_min;
        int    tm_hour;
        int    tm_mday;
        int    tm_mon;
        int    tm_year;
        int    tm_wday;
        int    tm_yday;
        int    tm_isdst;
        }tmx_t;
HcpyC_TIME(char *ap, double *bp)
{
double wk;
time_t wk2;
 tmx_t tim, *tim_p;
int i;
for(i=0; i<4; i++) {
*((short*)&wk+i) = *((short*)bp+i);
}
wk2 = (time_t)wk;
tim_p =  localtime(&wk2) ;
memcpy(&tim,tim_p,sizeof(tmx_t));
sprintf( ap, "%02d-%02d-%04d %02d:%02d:%02d",
tim.tm_mday, tim.tm_mon+1, tim.tm_year + 1900,
tim.tm_hour, tim.tm_min, tim.tm_sec );
}
HcpyCDATE_D(char *ap, double *bp)
{
double wk;
time_t wk2;
 tmx_t tim, *tim_p;
int i;
for(i=0; i<4; i++) {
*((short*)&wk+i) = *((short*)bp+i);
}
wk2 = (time_t)wk;
   tim_p =  localtime(&wk2) ;
       memcpy(&tim,tim_p,sizeof(tmx_t));
sprintf( ap, "%02d-%02d-%04d",
tim.tm_mday, tim.tm_mon+1, tim.tm_year + 1900 );
}
HcpyDRATE_I(double *ap, long *bp)
{
long wk;
int i;
for(i=0; i<2; i++) {
*((short*)&wk+i) = *((short*)bp+i);
}
*ap = ((double)wk) ∕ 100;
}
tput_orderstatus()
{
orderstat_trans *tiOS;
int rtn;
int w_id;
int tx_type;
w_id = iOS->w_id;
tx_type = TX_ORDERSTATUS;
tiOS = ( orderstat_trans *)twbuf;
```

```
tiOS->tx_type = TX_ORDERSTATUS;
tiOS->C_R = 0;
HcpyH_I(&tiOS->w_id,   &iOS->w_id);
HcpyH_I(&tiOS->d_id,   &iOS->d_id);
HcpyI_I(&tiOS->c_id,   &iOS->c_id);
strcpy( tiOS->c_last, iOS->c_last);
rtn = Fchg( (struct Fbfr *)trans_b, FML_TERM, 0, (char *)&w_id, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_TRAN, 0, (char *)&tx_type, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_DATA, 0, (char *)tiOS,
( FLDLEN )sizeof(
orderstat_trans ) );
}
tget_orderstatus()
{
orderstat_trans *toOS;
struct ord_itm_struct *sol;
int i;
toOS = ( orderstat_trans *)Ffind( (struct Fbfr *)trans_b, FML_DATA, 0, NULL );
if( toOS->C_R != 1 ){
sqlerror( TX_ORDERSTATUS, ( char * )toOS );
}
;;;i++;;
HcpyI_I(&oOS->c_id,       &toOS->c_id);
strcpy( oOS->c_last,      toOS->c_last);
strcpy( oOS->c_first,     toOS->c_first);
strcpy( oOS->c_middle,    toOS->c_middle);
HcpyD_DCENT(&oOS->c_balance, &toOS->c_balance);
HcpyI_I(&oOS->o_id,       &toOS->o_id);
HcpyC_TIME( oOS->o_entry_d,  &toOS->o_entry_d);
if( toOS->o_carrier_id == INTNULL ) {
oOS->o_carrier_id = 0;
} else {
HcpyI_H(&oOS->o_carrier_id,  &toOS->o_carrier_id);
}
HcpyI_H(&oOS->ol_cnt,       &toOS->o_ol_cnt);
for (i = 0; i < oOS->ol_cnt; i++) {
sol = &oOS->s_ol[i];
if (toOS->ol_supply_w_id[i] > 0) {
HcpyI_H(&sol->ol_supply_w_id, &toOS->ol_supply_w_id[i]);
HcpyI_I(&sol->ol_i_id,        &toOS->ol_i_id[i]);
HcpyI_H(&sol->ol_quantity,    &toOS->ol_quantity[i]);
HcpyD_ICENT(&sol->ol_amount,  &toOS->ol_amount[i]);
if( oOS->o_carrier_id == 0 ) {
strcpy(sol->ol_delivery_d,"99-99-9999");
} else {
HcpyCDATE_D( sol->ol_delivery_d,  &toOS->ol_delivery_d[i]);
}
}
}
}
tput_neworder()
{
neworder_trans *tiNO;
struct no_itm_struct *ol_ptr;
int rtn;
int w_id;
int tx_type;
int i;
w_id = iNO->w_id;
tx_type = TX_NEWORDER;
tiNO = (neworder_trans *)twbuf;
tiNO->tx_type = TX_NEWORDER;
tiNO->C_R = 0;
```

```
HcpyH_I(&tiNO->w_id,     &iNO->w_id);
HcpyH_I(&tiNO->d_id,     &iNO->d_id);
HcpyI(&tiNO->c_id,     &iNO->c_id);
for (i = 0; i < iNO->o_ol_cnt; i++) {
ol_ptr = &iNO->o_ol[i];
HcpyH_I(&tiNO->ol_supply_w_id[i],    &ol_ptr->ol_supply_w_id);
HcpyI(&tiNO->ol_i_id[i],          &ol_ptr->ol_i_id);
HcpyH_I(&tiNO->ol_quantity[i],      &ol_ptr->ol_quantity);
}
HcpyH_I(&tiNO->o_ol_cnt, &iNO->o_ol_cnt);
rtn = Fchg( (struct Fbfr *)trans_b, FML_TERM, 0, (char *)&w_id, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_TRAN, 0, (char *)&tx_type, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_DATA, 0, (char *)tiNO,
( FLDLEN )sizeof(
neworder_trans ) );
}
tget_neworder()
{
neworder_trans *toNO;
struct no_itm_struct *ool;
int i;
toNO = ( neworder_trans *)Ffind( (struct Fbfr *)trans_b, FML_DATA, 0,
NULL );
if( toNO->C_R == 0 ) {
  strcpy(oNO->status, "Transaction error occured");
sqlerror( TX_NEWORDER, ( char * )toNO );
    return(1);
} else if ( toNO->C_R == 2 ) {
strcpy(oNO->status, "Item number is not valid");
} else
{
  strcpy(oNO->status, blank_mesg);
}
HcpyC_TIME( oNO->o_entry_d,   &toNO->o_entry_d);
strcpy(  oNO->c_last,        toNO->c_last);
strcpy(  oNO->c_credit,       toNO->c_credit);
HcpyDRATE_I(&oNO->c_discount, &toNO->c_discount);
HcpyI_I(&oNO->o_id,          &toNO->o_id);
HcpyI_H(&oNO->o_ol_cnt,       &toNO->o_ol_cnt);
HcpyDRATE_I(&oNO->w_tax,      &toNO->w_tax);
HcpyDRATE_I(&oNO->d_tax,      &toNO->d_tax);
for (i = 0; i < oNO->o_ol_cnt; i++) {
ool = &oNO->o_ol[i];
HcpyD_ICENT(&ool->i_price, &toNO->i_price[i]);
strcpy(  ool->i_name,      &toNO->i_name[i]);
HcpyI_H(&ool->ol_quantity, &toNO->ol_quantity[i]);
HcpyI_I(&ool->s_quantity,  &toNO->s_quantity[i]);
memcpy(  ool->brand,       &toNO->brand_generic[i],1);
ool->brand[1] = '\0';
HcpyD_ICENT(&ool->ol_amount,  &toNO->ol_amount[i]);
}
HcpyD_ICENT(&oNO->total,&toNO->total_amount);
return(0);
}
tput_payment()
{
payment_trans *tiPT;
int rtn;
int w_id;
int tx_type;
w_id = iPT->w_id;
tx_type = TX_PAYMENT;
tiPT = (payment_trans *)twbuf;
```

```
tiPT->tx_type = TX_PAYMENT;
tiPT->C_R = 0;
HcpyH_I(&tiPT->w_id,        &iPT->w_id);
HcpyH_I(&tiPT->d_id,        &iPT->d_id);
HcpyH_I(&tiPT->c_w_id,      &iPT->c_w_id);
HcpyH_I(&tiPT->c_d_id,      &iPT->c_d_id);
HcpyICENT_D(&tiPT->h_amount, &iPT->h_amount);
HcpyI(&tiPT->c_id,        &iPT->c_id);
strcpy( tiPT->c_last,     iPT->c_last);
rtn = Fchg( (struct Fbfr *)trans_b, FML_TERM, 0, (char *)&w_id, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_TRAN, 0, (char *)&tx_type, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_DATA, 0, (char *)tiPT,
( FLDLEN )sizeof(
payment_trans ) );
}
tget_payment()
{
payment_trans *toPT;
int i;
toPT = ( payment_trans *)Ffind( (struct Fbfr *)trans_b, FML_DATA, 0, NULL
);
if( toPT->C_R != 1 ) {
sqlerror( TX_PAYMENT, ( char * )toPT );
}
HcpyC_TIME( oPT->h_date,       &toPT->h_date);
strcpy(  oPT->w_street_1,      toPT->w_street_1);
strcpy(  oPT->w_street_2,      toPT->w_street_2);
strcpy(  oPT->w_city,          toPT->w_city);
strcpy(  oPT->w_state,         toPT->w_state);
strncpy(  oPT->w_zip,          toPT->w_zip,10);
strcpy(  oPT->d_street_1,      toPT->d_street_1);
strcpy(  oPT->d_street_2,      toPT->d_street_2);
strcpy(  oPT->d_city,          toPT->d_city);
strcpy(  oPT->d_state,         toPT->d_state);
strncpy(  oPT->d_zip,          toPT->d_zip,10);
HcpyI_I(&oPT->c_id,            &toPT->c_id);
strcpy(  oPT->c_last,        toPT->c_last);
strcpy(  oPT->c_first,       toPT->c_first);
strcpy(  oPT->c_middle,        toPT->c_middle);
strcpy(  oPT->c_street_1,      toPT->c_street_1);
strcpy(  oPT->c_street_2,      toPT->c_street_2);
strcpy(  oPT->c_city,          toPT->c_city);
strcpy(  oPT->c_state,         toPT->c_state);
strcpy(  oPT->c_zip,           toPT->c_zip);
HcpyCDATE_D( oPT->c_since,      &toPT->c_since);
strcpy(  oPT->c_credit,        toPT->c_credit);
HcpyDRATE_I(&oPT->c_discount,   &toPT->c_discount);
strcpy(  oPT->c_phone,          toPT->c_phone);
HcpyD_DCENT(&oPT->c_balance,    &toPT->c_balance);
HcpyD_DCENT(&oPT->c_credit_lim, &toPT->c_credit_lim);
if( toPT->c_data[0] == NULL ) {
oPT->c_data_1[0] = '\0';
oPT->c_data_2[0] = '\0';
oPT->c_data_3[0] = '\0';
oPT->c_data_4[0] = '\0';
} else {
i = strlen(&toPT->c_data);
if( i > 0 ) {
oPT->c_data_1[0] = '\0';
strncpy( oPT->c_data_1, &toPT->c_data,50);
oPT->c_data_1[50] = '\0';
oPT->c_data_2[0] = '\0';
if( i > 50 ) {
```

```
strncpy( oPT->c_data_2, &toPT->c_data[50],50);
oPT->c_data_2[50] = '\0';
oPT->c_data_3[0] = '\0';
if( i > 100 ) {
strncpy( oPT->c_data_3, &toPT->c_data[100],50);
oPT->c_data_3[50] = '\0';
oPT->c_data_4[0] = '\0';
if( i > 150 ) {
memcpy( oPT->c_data_4, &toPT->c_data[150],50);
oPT->c_data_4[50] = '\0';
}
}
}
}
}
}
tput_delivery()
{
delivery_trans *tiDY;
int rtn;
int w_id;
int tx_type;
w_id = iDY->w_id;
tx_type = TX_DELIVERY;
tiDY = (delivery_trans *)twbuf;
tiDY->tx_type = TX_DELIVERY;
tiDY->C_R = 0;
HcpyH_I(&tiDY->w_id,        &iDY->w_id);
HcpyH_I(&tiDY->o_carrier_id, &iDY->o_carrier_id);
HcpyI_I(&tiDY->startsec,     &iDY->queue_time);
tiDY->startusec = 0;
rtn = Fchg( (struct Fbfr *)trans_b, FML_TERM, 0, (char *)&w_id, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_TRAN, 0, (char *)&tx_type, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_DATA, 0, (char *)tiDY,
( FLDLEN )sizeof(
delivery_trans ) );
}
tput_stocklevel()
{
stocklvl_trans *tiSL;
int rtn;
int w_id;
int tx_type;
w_id = iSL->w_id;
tx_type = TX_STOCKLEVEL;
tiSL = (stocklvl_trans *)twbuf;
tiSL->tx_type = TX_STOCKLEVEL;
tiSL->C_R = 0;
HcpyH_I(&tiSL->w_id,      &iSL->w_id);
HcpyH_I(&tiSL->d_id,      &iSL->d_id);
HcpyI_I(&tiSL->threshold, &iSL->threshold);
rtn = Fchg( (struct Fbfr *)trans_b, FML_TERM, 0, (char *)&w_id, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_TRAN, 0, (char *)&tx_type, 0 );
rtn = Fchg( (struct Fbfr *)trans_b, FML_DATA, 0, (char *)tiSL,
( FLDLEN )sizeof(
stocklvl_trans ) );
}
tget_stocklevel()
{
stocklvl_trans *toSL;
toSL = ( stocklvl_trans *)Ffind( (struct Fbfr *)trans_b, FML_DATA, 0,
NULL );
if( toSL->C_R != 1 ) {
```

```
sqlerror( TX_STOCKLEVEL, ( char * )toSL );
}
HcpyI_I(&oSL->low_stock,         &toSL->low_stock);
}
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#defineFML_TERM((FLDID)8193)
#defineFML_TRAN((FLDID)8194)
#defineFML_DATA((FLDID)49155)
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#define  DIST_PER_WARE 10
EXEC SQL BEGIN DECLARE SECTION;
short w_id;
char w_name[11];
char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
char w_zip[10];
long w_tax;
double w_ytd;
short d_id;
char d_name[11];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
long d_tax;
long d_ytd;
long d_next_o_id;
long c_id;
short c_d_id;
short c_w_id;
char c_first[17];
char c_middle[3];
char c_last[17];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];
char c_since[15];
char c_credit[3];
double c_credit_lim;
long c_discount;
double c_balance;
double c_ytd_payment;
short c_payment_cnt;
char c_data[501];
char h_date[15];
long h_amount;
char h_data[25];
long no_o_id;
long o_id;
char o_entry_d[15];
short o_carrier_id;
short o_ol_cnt;
short o_all_local;
```

```
long ol_number;                                    char   d_street_2[21];
long ol_i_id;                                      char   d_city[21];
short ol_supply_w_id;                              char   d_state[3];
char ol_delivery_d[15];                            char   d_zip[10];
short ol_quantity;                                 int    c_id;
long ol_amount;                                    short  c_d_id;
char ol_dist_info[25];                             short  c_w_id;
long s_quantity;                                   char   c_first[17];
char s_dist_01[25];                                char   c_middle[3];
char s_dist_02[25];                                char   c_last[17];
char s_dist_03[25];                                char   c_street_1[21];
char s_dist_04[25];                                char   c_street_2[21];
char s_dist_05[25];                                char   c_city[21];
char s_dist_06[25];                                char   c_state[3];
char s_dist_07[25];                                char   c_zip[10];
char s_dist_08[25];                                char   c_phone[17];
char s_dist_09[25];                                double c_since;
char s_dist_10[25];                                char   c_credit[3];
double s_ytd;                                      double c_credit_lim;
long s_order_cnt;                                  long   c_discount;
long s_remote_cnt;                                 double c_balance;
char s_data[51];                                   char   c_data[501];
long i_priceh;                                     double h_date;
char i_datah[51];                                  long   h_amount;
char i_nameh[25];                              } payment_trans;
EXEC SQL END DECLARE SECTION;                  typedef struct {
/*                                                 int    tx_type;
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc    int    C_R;
 */                                                int    errorpos;
typedef struct {                                   int    sqlstate;
      int    tx_type;                              short  w_id;
      int    C_R;                                  short  d_id;
      int    errorpos;                             int    c_id;
      int    sqlstate;                             char   c_first[17];
      short  w_id;                                 char   c_middle[3];
      short  d_id;                                 char   c_last[17];
      short  o_carrier_id;                         double c_balance;
      long   startsec;                             long   o_id;
      long   startusec;                            double o_entry_d;
} delivery_trans;                                  short  o_carrier_id;
typedef struct {                                   short  o_ol_cnt;
      int    tx_type;                              long   ol_i_id[15];
      int    C_R;                                  short  ol_supply_w_id[15];
      int    errorpos;                             double ol_delivery_d[15];
      int    sqlstate;                             short  ol_quantity[15];
      long   threshold;                            long   ol_amount[15];
      long   low_stock;                        } orderstat_trans;
      short  w_id;                           typedef struct {
      short  d_id;                                 int    tx_type;
} stocklvl_trans;                                  int    C_R;
typedef struct {                                   int    errorpos;
      int    tx_type;                              int    sqlstate;
      int    C_R;                                  char   brand_generic[15];
      int    errorpos;                             long   i_price[15];
      int    sqlstate;                             char   i_name[15][25];
      short  w_id;                                 long   total_amount;
      char   w_street_1[21];                       short  w_id;
      char   w_street_2[21];                       long   w_tax;
      char   w_city[21];                           short  d_id;
      char   w_state[3];                           long   d_tax;
      char   w_zip[10];                            int    c_id;
      short  d_id;                                 char   c_last[17];
      char   d_street_1[21];                       char   c_credit[3];
```

```
        long    c_discount;                                 long    o_id;
        long    o_id;                                       double  o_entry_d;
        double  o_entry_d;                                  short   o_carrier_id;
        short   o_ol_cnt;                                   short   o_ol_cnt;
        long    ol_i_id[15];                                short   o_all_local;
        short   ol_supply_w_id[15];                         long    ol_number;
        short   ol_quantity[15];                            long    ol_i_id[15];
        long    ol_amount[15];                              short   ol_supply_w_id[15];
        long    s_quantity[15];                             double  ol_delivery_d[15];
} neworder_trans;                                           short   ol_quantity[15];
#if 0                                                       long    ol_amount[15];
typedef struct {                                            char    ol_dist_info[24];
        int     tx_type;                                    long    s_quantity[15];
        int     C_R;                                        char    s_dist_01[24];
        long    threshold;                                  char    s_dist_02[24];
        long    low_stock;                                  char    s_dist_03[24];
        char    brand_generic[15];                          char    s_dist_04[24];
        long    i_price[15];                                char    s_dist_05[24];
        char    i_name[15][25];                             char    s_dist_06[24];
        long    total_amount;                               char    s_dist_07[24];
        double  pl_delivery_d[15];                          char    s_dist_08[24];
        short   w_id;                                       char    s_dist_09[24];
        char    w_name[11];                                 char    s_dist_10[24];
        char    w_street_1[21];                             double  s_ytd;
        char    w_street_2[21];                             long    s_order_cnt;
        char    w_city[21];                                 long    s_remote_cnt;
        char    w_state[3];                                 char    s_data[51];
        char    w_zip[10];                          } trans_buf;
        long    w_tax;                              main()
        double  w_ytd;                              {
        short   d_id;                                       printf( "%d %d %d %d %d %d\n",
        char    d_name[11];                                         sizeof( delivery_trans ),
        char    d_street_1[21];                                     sizeof( stocklvl_trans ),
        char    d_street_2[21];                                     sizeof( payment_trans ),
        char    d_city[21];                                         sizeof( orderstat_trans ),
        char    d_state[3];                                         sizeof( neworder_trans ),
        char    d_zip[10];                                          sizeof( trans_buf ) );
        long    d_tax;                                      return 0;
        long    d_next_o_id;                        }
        int     c_id;                               #endif
        short   c_d_id;                             /*
        short   c_w_id;                              * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
        char    c_first[17];                         */
        char    c_middle[3];                           typedef struct{
        char    c_last[17];                                 long    ol_o_id;
        char    c_street_1[21];                             short   ol_d_id;
        char    c_street_2[21];                             short   ol_w_id;
        char    c_city[21];                                 long    ol_number;
        char    c_state[3];                                 long    ol_i_id;
        char    c_zip[10];                                  short   ol_supply_w_id;
        char    c_phone[17];                                short   ol_quantity;
        double  c_since;                                    long    ol_amount;
        char    c_credit[3];                                char    ol_dist_info[25];
        double  c_credit_lim;                               char    dummy2[3];
        long    c_discount;                            }lnk_ol;
        double  c_balance;                          /*
        double  c_ytd_payment;                       * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
        short   c_payment_cnt;                       */
        char    c_data[501];                        EXEC SQL BEGIN DECLARE SECTION;
        double  h_date;                             char        state[6];
        long    h_amount;                           char        sqlmsg[257];
        char    h_data[25];                         short       sqlmsg_ind;
        long    no_o_id;                            int         errorpos;
```

short	w_name_ind;	short	ol_i_id10_ind;
short	w_street_1_ind;	short	ol_i_id11_ind;
short	w_street_2_ind;	short	ol_i_id12_ind;
short	w_city_ind;	short	ol_i_id13_ind;
short	w_state_ind;	short	ol_i_id14_ind;
short	w_zip_ind;	short	ol_i_id15_ind;
short	w_tax_ind;	short	ol_supply_w_id1;
short	d_id_ind;	short	ol_supply_w_id2;
short	d_name_ind;	short	ol_supply_w_id3;
short	d_street_1_ind;	short	ol_supply_w_id4;
short	d_street_2_ind;	short	ol_supply_w_id5;
short	d_city_ind;	short	ol_supply_w_id6;
short	d_state_ind;	short	ol_supply_w_id7;
short	d_zip_ind;	short	ol_supply_w_id8;
short	d_tax_ind;	short	ol_supply_w_id9;
short	c_id_ind;	short	ol_supply_w_id10;
short	c_first_ind;	short	ol_supply_w_id11;
short	c_middle_ind;	short	ol_supply_w_id12;
short	c_last_ind;	short	ol_supply_w_id13;
short	c_street_1_ind;	short	ol_supply_w_id14;
short	c_street_2_ind;	short	ol_supply_w_id15;
short	c_city_ind;	short	ol_supply_w_id1_ind;
short	c_state_ind;	short	ol_supply_w_id2_ind;
short	c_zip_ind;	short	ol_supply_w_id3_ind;
short	c_phone_ind;	short	ol_supply_w_id4_ind;
short	c_credit_ind;	short	ol_supply_w_id5_ind;
short	c_credit_lim_ind;	short	ol_supply_w_id6_ind;
short	c_discount_ind;	short	ol_supply_w_id7_ind;
short	c_balance_ind;	short	ol_supply_w_id8_ind;
short	c_ytd_payment_ind;	short	ol_supply_w_id9_ind;
short	c_payment_cnt_ind;	short	ol_supply_w_id10_ind;
short	c_since_ind;	short	ol_supply_w_id11_ind;
varchar	c_datax[501];	short	ol_supply_w_id12_ind;
short	c_data_ind ;	short	ol_supply_w_id13_ind;
short	o_id_ind;	short	ol_supply_w_id14_ind;
short	o_entry_d_ind;	short	ol_supply_w_id15_ind;
short	o_carrier_id_ind;	short	ol_quantity1;
short	o_all_local_ind;	short	ol_quantity2;
short	no_o_id_ind;	short	ol_quantity3;
long	ol_i_id1;	short	ol_quantity4;
long	ol_i_id2;	short	ol_quantity5;
long	ol_i_id3;	short	ol_quantity6;
long	ol_i_id4;	short	ol_quantity7;
long	ol_i_id5;	short	ol_quantity8;
long	ol_i_id6;	short	ol_quantity9;
long	ol_i_id7;	short	ol_quantity10;
long	ol_i_id8;	short	ol_quantity11;
long	ol_i_id9;	short	ol_quantity12;
long	ol_i_id10;	short	ol_quantity13;
long	ol_i_id11;	short	ol_quantity14;
long	ol_i_id12;	short	ol_quantity15;
long	ol_i_id13;	short	ol_quantity1_ind;
long	ol_i_id14;	short	ol_quantity2_ind;
long	ol_i_id15;	short	ol_quantity3_ind;
short	ol_i_id1_ind;	short	ol_quantity4_ind;
short	ol_i_id2_ind;	short	ol_quantity5_ind;
short	ol_i_id3_ind;	short	ol_quantity6_ind;
short	ol_i_id4_ind;	short	ol_quantity7_ind;
short	ol_i_id5_ind;	short	ol_quantity8_ind;
short	ol_i_id6_ind;	short	ol_quantity9_ind;
short	ol_i_id7_ind;	short	ol_quantity10_ind;
short	ol_i_id8_ind;	short	ol_quantity11_ind;
short	ol_i_id9_ind;	short	ol_quantity12_ind;

```
short      ol_quantity13_ind;          long      s_quantity1;
short      ol_quantity14_ind;          long      s_quantity2;
short      ol_quantity15_ind;          long      s_quantity3;
int        ol_amount1;                 long      s_quantity4;
int        ol_amount2;                 long      s_quantity5;
int        ol_amount3;                 long      s_quantity6;
int        ol_amount4;                 long      s_quantity7;
int        ol_amount5;                 long      s_quantity8;
int        ol_amount6;                 long      s_quantity9;
int        ol_amount7;                 long      s_quantity10;
int        ol_amount8;                 long      s_quantity11;
int        ol_amount9;                 long      s_quantity12;
int        ol_amount10;                long      s_quantity13;
int        ol_amount11;                long      s_quantity14;
int        ol_amount12;                long      s_quantity15;
int        ol_amount13;                short     s_quantity1_ind;
int        ol_amount14;                short     s_quantity2_ind;
int        ol_amount15;                short     s_quantity3_ind;
short      ol_amount1_ind;             short     s_quantity4_ind;
short      ol_amount2_ind;             short     s_quantity5_ind;
short      ol_amount3_ind;             short     s_quantity6_ind;
short      ol_amount4_ind;             short     s_quantity7_ind;
short      ol_amount5_ind;             short     s_quantity8_ind;
short      ol_amount6_ind;             short     s_quantity9_ind;
short      ol_amount7_ind;             short     s_quantity10_ind;
short      ol_amount8_ind;             short     s_quantity11_ind;
short      ol_amount9_ind;             short     s_quantity12_ind;
short      ol_amount10_ind;            short     s_quantity13_ind;
short      ol_amount11_ind;            short     s_quantity14_ind;
short      ol_amount12_ind;            short     s_quantity15_ind;
short      ol_amount13_ind;            char      s_dist1[25];
short      ol_amount14_ind;            char      s_dist2[25];
short      ol_amount15_ind;            char      s_dist3[25];
char       ol_delivery_d1[14];         char      s_dist4[25];
char       ol_delivery_d2[14];         char      s_dist5[25];
char       ol_delivery_d3[14];         char      s_dist6[25];
char       ol_delivery_d4[14];         char      s_dist7[25];
char       ol_delivery_d5[14];         char      s_dist8[25];
char       ol_delivery_d6[14];         char      s_dist9[25];
char       ol_delivery_d7[14];         char      s_dist10[25];
char       ol_delivery_d8[14];         char      s_dist11[25];
char       ol_delivery_d9[14];         char      s_dist12[25];
char       ol_delivery_d10[14];        char      s_dist13[25];
char       ol_delivery_d11[14];        char      s_dist14[25];
char       ol_delivery_d12[14];        char      s_dist15[25];
char       ol_delivery_d13[14];        short     s_dist1_ind;
char       ol_delivery_d14[14];        short     s_dist2_ind;
char       ol_delivery_d15[14];        short     s_dist3_ind;
short      ol_delivery_d1_ind;         short     s_dist4_ind;
short      ol_delivery_d2_ind;         short     s_dist5_ind;
short      ol_delivery_d3_ind;         short     s_dist6_ind;
short      ol_delivery_d4_ind;         short     s_dist7_ind;
short      ol_delivery_d5_ind;         short     s_dist8_ind;
short      ol_delivery_d6_ind;         short     s_dist9_ind;
short      ol_delivery_d7_ind;         short     s_dist10_ind;
short      ol_delivery_d8_ind;         short     s_dist11_ind;
short      ol_delivery_d9_ind;         short     s_dist12_ind;
short      ol_delivery_d10_ind;        short     s_dist13_ind;
short      ol_delivery_d11_ind;        short     s_dist14_ind;
short      ol_delivery_d12_ind;        short     s_dist15_ind;
short      ol_delivery_d13_ind;        long      i_priceh1;
short      ol_delivery_d14_ind;        long      i_priceh2;
short      ol_delivery_d15_ind;        long      i_priceh3;
```

A ≡

```
long        i_priceh4;                          char        i_datah7[51];
long        i_priceh5;                          char        i_datah8[51];
long        i_priceh6;                          char        i_datah9[51];
long        i_priceh7;                          char        i_datah10[51];
long        i_priceh8;                          char        i_datah11[51];
long        i_priceh9;                          char        i_datah12[51];
long        i_priceh10;                         char        i_datah13[51];
long        i_priceh11;                         char        i_datah14[51];
long        i_priceh12;                         char        i_datah15[51];
long        i_priceh13;                         short       i_datah1_ind;
long        i_priceh14;                         short       i_datah2_ind;
long        i_priceh15;                         short       i_datah3_ind;
short       i_priceh1_ind;                      short       i_datah4_ind;
short       i_priceh2_ind;                      short       i_datah5_ind;
short       i_priceh3_ind;                      short       i_datah6_ind;
short       i_priceh4_ind;                      short       i_datah7_ind;
short       i_priceh5_ind;                      short       i_datah8_ind;
short       i_priceh6_ind;                      short       i_datah9_ind;
short       i_priceh7_ind;                      short       i_datah10_ind;
short       i_priceh8_ind;                      short       i_datah11_ind;
short       i_priceh9_ind;                      short       i_datah12_ind;
short       i_priceh10_ind;                     short       i_datah13_ind;
short       i_priceh11_ind;                     short       i_datah14_ind;
short       i_priceh12_ind;                     short       i_datah15_ind;
short       i_priceh13_ind;                     int         result_o_id1;
short       i_priceh14_ind;                     int         result_o_id2;
short       i_priceh15_ind;                     int         result_o_id3;
char        i_nameh1[25];                        int         result_o_id4;
char        i_nameh2[25];                        int         result_o_id5;
char        i_nameh3[25];                        int         result_o_id6;
char        i_nameh4[25];                        int         result_o_id7;
char        i_nameh5[25];                        int         result_o_id8;
char        i_nameh6[25];                        int         result_o_id9;
char        i_nameh7[25];                        int         result_o_id10;
char        i_nameh8[25];                        int         result_o_id11;
char        i_nameh9[25];                        int         result_o_id12;
char        i_nameh10[25];                       int         result_o_id13;
char        i_nameh11[25];                       int         result_o_id14;
char        i_nameh12[25];                       int         result_o_id15;
char        i_nameh13[25];                       short       result_o_id1_ind;
char        i_nameh14[25];                       short       result_o_id2_ind;
char        i_nameh15[25];                       short       result_o_id3_ind;
short       i_nameh1_ind;                        short       result_o_id4_ind;
short       i_nameh2_ind;                        short       result_o_id5_ind;
short       i_nameh3_ind;                        short       result_o_id6_ind;
short       i_nameh4_ind;                        short       result_o_id7_ind;
short       i_nameh5_ind;                        short       result_o_id8_ind;
short       i_nameh6_ind;                        short       result_o_id9_ind;
short       i_nameh7_ind;                        short       result_o_id10_ind;
short       i_nameh8_ind;                        short       result_o_id11_ind;
short       i_nameh9_ind;                        short       result_o_id12_ind;
short       i_nameh10_ind;                       short       result_o_id13_ind;
short       i_nameh11_ind;                       short       result_o_id14_ind;
short       i_nameh12_ind;                       short       result_o_id15_ind;
short       i_nameh13_ind;                       short       notfound;
short       i_nameh14_ind;                       short       notfound_ind;
short       i_nameh15_ind;                       short       item_notfound;
char        i_datah1[51];                        short       item_notfound_ind;
char        i_datah2[51];                        short       low_stock_ind;
char        i_datah3[51];                        EXEC SQL END DECLARE SECTION;
char        i_datah4[51];                        long    *ol_i_id_str[]      = { (long  *)&ol_i_id1      ,
char        i_datah5[51];                                                        (long  *)&ol_i_id2      ,
char        i_datah6[51];                                                        (long  *)&ol_i_id3      ,
```

Appendix A: Application Code   61

```
                    (long  *)&ol_i_id4       ,                              (char  *)&ol_delivery_d3    ,
                    (long  *)&ol_i_id5       ,                              (char  *)&ol_delivery_d4    ,
                    (long  *)&ol_i_id6       ,                              (char  *)&ol_delivery_d5    ,
                    (long  *)&ol_i_id7       ,                              (char  *)&ol_delivery_d6    ,
                    (long  *)&ol_i_id8       ,                              (char  *)&ol_delivery_d7    ,
                    (long  *)&ol_i_id9       ,                              (char  *)&ol_delivery_d8    ,
                    (long  *)&ol_i_id10      ,                              (char  *)&ol_delivery_d9    ,
                    (long  *)&ol_i_id11      ,                              (char  *)&ol_delivery_d10   ,
                    (long  *)&ol_i_id12      ,                              (char  *)&ol_delivery_d11   ,
                    (long  *)&ol_i_id13      ,                              (char  *)&ol_delivery_d12   ,
                    (long  *)&ol_i_id14      ,                              (char  *)&ol_delivery_d13   ,
                    (long  *)&ol_i_id15      ,                              (char  *)&ol_delivery_d14   ,
                    NULL};                                                  (char  *)&ol_delivery_d15   ,
short    *ol_supply_w_id_str[] = { (short *)&ol_supply_w_id1 ,              NULL};
                    (short *)&ol_supply_w_id2 ,         long    *s_quantity_str[]    = { (long  *)&s_quantity1     ,
                    (short *)&ol_supply_w_id3 ,                             (long  *)&s_quantity2      ,
                    (short *)&ol_supply_w_id4 ,                             (long  *)&s_quantity3      ,
                    (short *)&ol_supply_w_id5 ,                             (long  *)&s_quantity4      ,
                    (short *)&ol_supply_w_id6 ,                             (long  *)&s_quantity5      ,
                    (short *)&ol_supply_w_id7 ,                             (long  *)&s_quantity6      ,
                    (short *)&ol_supply_w_id8 ,                             (long  *)&s_quantity7      ,
                    (short *)&ol_supply_w_id9 ,                             (long  *)&s_quantity8      ,
                    (short *)&ol_supply_w_id10,                             (long  *)&s_quantity9      ,
                    (short *)&ol_supply_w_id11,                             (long  *)&s_quantity10     ,
                    (short *)&ol_supply_w_id12,                             (long  *)&s_quantity11     ,
                    (short *)&ol_supply_w_id13,                             (long  *)&s_quantity12     ,
                    (short *)&ol_supply_w_id14,                             (long  *)&s_quantity13     ,
                    (short *)&ol_supply_w_id15,                             (long  *)&s_quantity14     ,
                    NULL};                                                  (long  *)&s_quantity15     ,
short    *ol_quantity_str[]    = { (short *)&ol_quantity1    ,              NULL};
                    (short *)&ol_quantity2    ,         char    *s_dist_str[]        = { (char  *)&(s_dist1[0])    ,
                    (short *)&ol_quantity3    ,                             (char  *)&s_dist2         ,
                    (short *)&ol_quantity4    ,                             (char  *)&s_dist3         ,
                    (short *)&ol_quantity5    ,                             (char  *)&s_dist4         ,
                    (short *)&ol_quantity6    ,                             (char  *)&s_dist5         ,
                    (short *)&ol_quantity7    ,                             (char  *)&s_dist6         ,
                    (short *)&ol_quantity8    ,                             (char  *)&s_dist7         ,
                    (short *)&ol_quantity9    ,                             (char  *)&s_dist8         ,
                    (short *)&ol_quantity10   ,                             (char  *)&s_dist9         ,
                    (short *)&ol_quantity11   ,                             (char  *)&s_dist10        ,
                    (short *)&ol_quantity12   ,                             (char  *)&s_dist11        ,
                    (short *)&ol_quantity13   ,                             (char  *)&s_dist12        ,
                    (short *)&ol_quantity14   ,                             (char  *)&s_dist13        ,
                    (short *)&ol_quantity15   ,                             (char  *)&s_dist14        ,
                    NULL};                                                  (char  *)&s_dist15        ,
int      *ol_amount_str[]      = { (int  *)&ol_amount1      ,              NULL};
                    (int  *)&ol_amount2       ,         long    *i_priceh_str[]      = { (long  *)&i_priceh1       ,
                    (int  *)&ol_amount3       ,                             (long  *)&i_priceh2       ,
                    (int  *)&ol_amount4       ,                             (long  *)&i_priceh3       ,
                    (int  *)&ol_amount5       ,                             (long  *)&i_priceh4       ,
                    (int  *)&ol_amount6       ,                             (long  *)&i_priceh5       ,
                    (int  *)&ol_amount7       ,                             (long  *)&i_priceh6       ,
                    (int  *)&ol_amount8       ,                             (long  *)&i_priceh7       ,
                    (int  *)&ol_amount9       ,                             (long  *)&i_priceh8       ,
                    (int  *)&ol_amount10      ,                             (long  *)&i_priceh9       ,
                    (int  *)&ol_amount11      ,                             (long  *)&i_priceh10      ,
                    (int  *)&ol_amount12      ,                             (long  *)&i_priceh11      ,
                    (int  *)&ol_amount13      ,                             (long  *)&i_priceh12      ,
                    (int  *)&ol_amount14      ,                             (long  *)&i_priceh13      ,
                    (int  *)&ol_amount15      ,                             (long  *)&i_priceh14      ,
                    NULL};                                                  (long  *)&i_priceh15      ,
char    *ol_delivery_d_str[]   = { (char  *)&(ol_delivery_d1[0]),          NULL};
                    (char  *)&ol_delivery_d2    ,        char    *i_nameh_str[]       = { (char  *)&(i_nameh1[0])   ,
```

```
                        (char  *)&i_nameh2      ,            typedef struct
                        (char  *)&i_nameh3      ,              {
                        (char  *)&i_nameh4      ,                short      sqllen         ;
                        (char  *)&i_nameh5      ,                struct
                        (char  *)&i_nameh6      ,                  {
                        (char  *)&i_nameh7      ,                   char    s_quantity[6]    ;
                        (char  *)&i_nameh8      ,                   char    s_dist[24]       ;
                        (char  *)&i_nameh9      ,                   char    s_data[50]       ;
                        (char  *)&i_nameh10     ,                   char    sapstop[1]       ;
                        (char  *)&i_nameh11     ,                  } sqlvar[15]             ;
                        (char  *)&i_nameh12     ,              } s_join_str               ;
                        (char  *)&i_nameh13     ,            typedef struct
                        (char  *)&i_nameh14     ,              {
                        (char  *)&i_nameh15     ,                short      sqllen         ;
                        NULL};                                  struct
char    *i_datah_str[]    = { (char  *)&(i_datah1[0])   ,          {
                        (char  *)&i_datah2     ,                   char    i_price[6]       ;
                        (char  *)&i_datah3     ,                   char    i_name[24]       ;
                        (char  *)&i_datah4     ,                   char    i_data[50]       ;
                        (char  *)&i_datah5     ,                   char    sapstop[1]       ;
                        (char  *)&i_datah6     ,                  } sqlvar[15]             ;
                        (char  *)&i_datah7     ,              } i_join_str               ;
                        (char  *)&i_datah8     ,            typedef struct
                        (char  *)&i_datah9     ,              {
                        (char  *)&i_datah10    ,                short      sqllen         ;
                        (char  *)&i_datah11    ,                struct
                        (char  *)&i_datah12    ,                  {
                        (char  *)&i_datah13    ,                   char    ol_i_id[7]       ;
                        (char  *)&i_datah14    ,                   char    ol_amount[8]     ;
                        (char  *)&i_datah15    ,                   char    ol_supply_w_id[4] ;
                        NULL};                                    char    ol_quantity[4]   ;
int     *result_o_id_str[]    = { (int  *)&result_o_id1   ,        char    ol_delivery_d[14] ;
                        (int  *)&result_o_id2  ,                   char    sapstop[1]       ;
                        (int  *)&result_o_id3  ,                  } sqlvar[15]             ;
                        (int  *)&result_o_id4  ,              } ol_join_str              ;
                        (int  *)&result_o_id5  ,            typedef struct
                        (int  *)&result_o_id6  ,              {
                        (int  *)&result_o_id7  ,                short      sqllen         ;
                        (int  *)&result_o_id8  ,                struct
                        (int  *)&result_o_id9  ,                  {
                        (int  *)&result_o_id10 ,                   char    ol_quantity[4]   ;
                        (int  *)&result_o_id11 ,                  } sqlvar[15]             ;
                        (int  *)&result_o_id12 ,              } ol_q_join_str             ;
                        (int  *)&result_o_id13 ,            typedef struct
                        (int  *)&result_o_id14 ,              {
                        (int  *)&result_o_id15 ,                short      sqllen         ;
                        NULL};                                  struct
EXEC SQL BEGIN DECLARE SECTION     ;                          {
varchar        s_join[1216]   ;                                 char    ol_supply_w_id[4] ;
short          s_join_ind     ;                                } sqlvar[15]             ;
varchar        i_join[1216]   ;                            } ol_s_join_str             ;
short          i_join_ind     ;                          typedef struct
varchar        ol_join[571]   ;                            {
short          ol_join_ind    ;                             short      sqllen         ;
varchar        ol_q_join[61]  ;                             struct
short          ol_q_join_ind  ;                               {
varchar        ol_s_join[61]  ;                              char    ol_i_id[7]       ;
short          ol_s_join_ind  ;                             } sqlvar[15]             ;
varchar        ol_i_join[106] ;                          } ol_i_join_str             ;
short          ol_i_join_ind  ;                          typedef struct
varchar        result_join[101] ;                          {
short          result_join_ind ;                            short      sqllen         ;
EXEC SQL END DECLARE SECTION       ;                        struct
```

```
    {
      char    result_o_id[9]    ;
      char    sapstop[1]        ;
    } sqlvar[10]                 ;
  } result_join_str             ;
/*
 * Copyright(c) 1995, 1996, 1997, 1998, 1999 by Sun Microsystems, Inc
 */
#define FFLUSH_OUT
#ifdef NO_SQL
#else
#define USE_SQL_MODE
#endif
#ifdef TAMESHI_OZA
short  o_ol_cnt_kari;
#endif
#ifdef NT
#include <windows.h>
#endif
#include <sys/types.h>
#include <time.h>
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#ifndef NT
#include <unistd.h>
#endif
#include <string.h>
#include "atmi.h"
#include "tmenv.h"
#include "bench2.h"
#ifdefUSE_FML
# include"fml.h"
# include"fldtbl.h"
#endif
#include "bench3.h"
#ifdef NT
#include <WTYPES.H>
#endif
#include <stdio.h>
extern void JMPCINT2(),JMPCINT3();
extern long OLINSETT(lnk_ol *a,short *b,char *c);
EXEC SQL INCLUDE bench1.h;
EXEC SQL INCLUDE stored.h;
#define INTNULL -32768
#define DP userlog
#define RDB_NORMAL 0
#ifdef NT
file://#define TIMES GetSystemTime(&systemtime);\
//              tp.tv_sec = ((systemtime.wYear - 1970) *365*24*3600)
\
//                              + ((systemtime.wMonth - 1)
*30*24*3600    \
//                              + ((systemtime.wDay - 1) *24*3600
\
//                              + (systemtime.wHour * 3600
\
//                              + (systemtime.wMinute * 60
\
//                              + (systemtime.wSecond );\
//              tp.tv_usec = systemtime.wMilliseconds * 1000;\
//              tv_st_sec=tp.tv_sec; tv_st_usec=tp.tv_usec;
#define TIMES GetSystemTime(&systemtime);\
```

```
            yDay = 0;\
            switch(systemtime.wMonth - 1){\
            case 11:yDay += 30;\
            case 10:yDay += 31;\
            case 9:     yDay += 30;\
            case 8:     yDay += 31;\
            case 7:     yDay += 31;\
            case 6:     yDay += 30;\
            case 5:     yDay += 31;\
            case 4:     yDay += 30;\
            case 3:     yDay += 31;\
            case 2:     \
                        (((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 100 != 0)) | | \
                        ((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 400 == 0)))\
                            ?(yDay += 29): (yDay += 28);\
            case 1:     yDay += 31;\
            default:break;\
            }\
            \
            tp.tv_sec = systemtime.wSecond +\
                        (systemtime.wMinute * 60) +\
                        ((systemtime.wHour) * 3600) +\
                        ((systemtime.wDay - 1) * 3600 * 24);\
            tp.tv_sec += (yDay * 3600 * 24);    \
            \
            yDiff = systemtime.wYear - 1970;\
            work_day = (yDiff / 4);\
            tp.tv_sec += ((yDiff * 365) + work_day) * 24 * 3600;\
            tp.tv_usec = systemtime.wMilliseconds;\
            tv_st_sec=tp.tv_sec; tv_st_usec=tp.tv_usec;
#else
#define TIMES Gettimeofday(&tp); tv_st_sec=tp.tv_sec; tv_st_usec=tp.tv_usec;
#endif
#ifdef NT
#define TIMEE(NUM) \
//      GetSystemTime(&systemtime);\
//          tp.tv_sec = ((systemtime.wYear - 1970) *365*24*3600) \
//              + ((systemtime.wMonth - 1) *30*24*3600     \
//              + ((systemtime.wDay - 1) *24*3600          \
//              + (systemtime.wHour * 3600                 \
//              + (systemtime.wMinute * 60                 \
//              + (systemtime.wSecond );\
            GetSystemTime(&systemtime);\
            yDay = 0;\
            switch(systemtime.wMonth - 1){\
            case 11:yDay += 30;\
            case 10:yDay += 31;\
            case 9:     yDay += 30;\
            case 8:     yDay += 31;\
            case 7:     yDay += 31;\
            case 6:     yDay += 30;\
            case 5:     yDay += 31;\
            case 4:     yDay += 30;\
            case 3:     yDay += 31;\
            case 2:     \
                (((systemtime.wYear % 4 == 0) && (systemtime.wYear % 100 !=
0)) | | \
                ((systemtime.wYear % 4 == 0) && (systemtime.wYear % 400 ==
0)))\
                        ?(yDay += 29): (yDay += 28);\
            case 1:     yDay += 31;\
```

```
        default:break;\
        }\
        \
        tp.tv_sec = systemtime.wSecond +         \
                    (systemtime.wMinute * 60) +\
                    ((systemtime.wHour) * 3600) +\
                    ((systemtime.wDay - 1) * 3600 * 24);\
        tp.tv_sec += (yDay * 3600 * 24);     \
        \
        yDiff = systemtime.wYear - 1970;\
        work_day = (yDiff / 4);\
        tp.tv_sec += ((yDiff * 365) + work_day) * 24 * 3600;\
        tp.tv_usec = systemtime.wMilliseconds;\
    time_sec=tp.tv_sec-tv_st_sec;\
    if(tp.tv_usec < tv_st_usec) \
    {    time_usec=1000-tv_st_usec+tp.tv_usec; time_sec=time_sec-1; } \
    else \
    time_usec=tp.tv_usec-tv_st_usec; \
    time_usec=time_sec*1000+time_usec; \
    if(NUM!=999) \
    {    fprintf(time_fd,"SQL_NUM = %d EACH_TIME=
%d\n",NUM,time_usec); \
        all_time(NUM,time_sec,time_usec); \
    } \
    else \
    {    fprintf(time_fd,"ALL_NUM = %d EACH_TIME= %d.%06d\n", \
                                NUM, time_sec, time_usec ); \
    }
#else
#define TIMEE(NUM) \
    Gettimeofday(&tp);\
    time_sec=tp.tv_sec-tv_st_sec;\
    if(tp.tv_usec < tv_st_usec) \
    {    time_usec=1000000-tv_st_usec+tp.tv_usec; time_sec=time_sec-1;
} \
    else \
    time_usec=tp.tv_usec-tv_st_usec; \
    time_usec=time_sec*1000000+time_usec; \
    if(NUM!=999) \
    {    fprintf(time_fd,"SQL_NUM = %d EACH_TIME=
%d\n",NUM,time_usec); \
        all_time(NUM,time_sec,time_usec); \
    } \
    else \
    {    fprintf(time_fd,"ALL_NUM = %d EACH_TIME= %d.%06d\n", \
                                NUM, time_sec, time_usec ); \
    }
#endif
#ifdef UXP_DS
#define Gettimeofday(a) gettimeofday(a)
#elif defined NT
file://#define Gettimeofday(a) GetSystemTime(&systemtime);\
//                          *a.tv_sec = ((systemtime.wYear - 1970)
*365*24*3600) \
//                              + ((systemtime.wMonth - 1)
*30*24*3600)    \
//                              + ((systemtime.wDay - 1)
*24*3600)       \
//                              + (systemtime.wHour *
3600)           \
//                              + (systemtime.wMinute *
60)          \
//                              + (systemtime.wSecond );\
//                              *a.tv_usec = systemtime.wMilliseconds *
1000;
#define Gettimeofday(a) GetSystemTime(&systemtime);\
                        yDay = 0;\
                        switch(systemtime.wMonth - 1){\
                        case 11:yDay += 30;\
                        case 10:yDay += 31;\
                        case 9:     yDay += 30;\
                        case 8:     yDay += 31;\
                        case 7:     yDay += 31;\
                        case 6:     yDay += 30;\
                        case 5:     yDay += 31;\
                        case 4:     yDay += 30;\
                        case 3:     yDay += 31;\
                        case 2:\
                            \
                        (((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 100 != 0)) ||\
                            ((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 400 == 0)))\
                                ?(yDay += 29): (yDay +=
28);\
                        case 1:     yDay += 31;\
                        default:break;\
                        }\
                        *a.tv_sec = systemtime.wSecond +\
                                (systemtime.wMinute * 60)
+                       \
                                ((systemtime.wHour) *
3600) +                 \
                                ((systemtime.wDay - 1) *
3600 * 24);        \
                        *a.tv_sec += (yDay * 3600 * 24);\
                        \
                        \
                        yDiff = systemtime.wYear - 1970;\
                        work_day = (yDiff / 4);\
                        *a.tv_sec += ((yDiff * 365) + work_day) * 24
* 3600;\
                        *a.tv_usec = systemtime.wMilliseconds;
#else
#define Gettimeofday(a) gettimeofday(a,0)
#endif
extern int  scanstring();
time_t  tttt;
time_t  t_wk;
char    tc_wk[26];
char    tc_s[15];
#ifdef USE_SQL_MODE
EXEC SQL BEGIN DECLARE SECTION;
short errorpos_ind;
#endif
int tmp_s_i_id;
int tmp_w_id;
int tmp_d_id;
long namecount;
long ol_total;
long low_stock;
long threshold;
int tmp_o_id;
char SQLSTATE[6];
int     t19,t18,t17,t16,t15,t14,t13,t12,t11,t10,t09,t08,t07,t06,t05,t04,t03,t02;
#ifdef USE_SQL_MODE
```

```
EXEC SQL END DECLARE SECTION;
#else
#define OLINSERT OLINSERT_nop
#define JMPCINT2 JMPCINT2_nop
#define JMPCINT3 JMPCINT3_nop
OLINSERT_nop(){}
JMPCINT2_nop(){}
JMPCINT3_nop(){}
#ifdef NT
#defineSQLWAIT_O   Sleep( 1 );
#defineSQLWAIT_N   Sleep( 1 );
#defineSQLWAIT_N_C Sleep( 1 );
#defineSQLWAIT_N_R Sleep( 2 );
#defineSQLWAIT_P   Sleep( 1 );
#defineSQLWAIT_D   Sleep( 5 );
#defineSQLWAIT_S   Sleep( 2 );
#else
#define SLEEP_MIN 10
#defineSQLWAIT_O   usleep(    10 * SLEEP_MIN);
#defineSQLWAIT_N   usleep(   100 * SLEEP_MIN);
#defineSQLWAIT_N_C usleep(    10 * SLEEP_MIN);
#defineSQLWAIT_N_R usleep(   200 * SLEEP_MIN);
#defineSQLWAIT_P   usleep(    20 * SLEEP_MIN);
#defineSQLWAIT_D   usleep(   500 * SLEEP_MIN);
#defineSQLWAIT_S   usleep(   200 * SLEEP_MIN);
#endif
#endif
neworder_trans  *bpn;
payment_trans   *bpp;
orderstat_trans *bpo;
delivery_trans  *bpd;
stocklvl_trans  *bps;
#ifdefUSE_FML
neworder_trans nbuf;
payment_trans   pbuf;
orderstat_trans obuf;
delivery_trans dbuf;
stocklvl_trans sbuf;
#endif
#ifdef NT
struct _SYSTEMTIME systemtime;
struct tp_tag{
        longtv_sec     ;
        long  tv_usec      ;
    } ;
struct tp_tag tp,tp_e;
#else
struct timeval tp,tp_e;
#endif
long tv_st_sec,tv_st_usec;
long time_sec,time_usec;
long result_o_id[10];
int number;
int douitu;
FILE *fd = 0;
FILE *t_fd = 0;
FILE *time_fd ;
FILE    *delivery_handle = NULL;
FILE    *fp;
    static FILE *tpsvrinit_fp = 0;
    static ctr = 0;
void    s_ymdhms()
{
```

```
struct tm       tim;
time(&t_wk)                  ;
tim = *( localtime( &t_wk ) ) ;
sprintf( tc_s, “%04d%02d%02d%02d%02d%02d”,
        tim.tm_year + 1900, tim.tm_mon+1, tim.tm_mday,
        tim.tm_hour, tim.tm_min, tim.tm_sec ) ;
tc_s[14] = 0 ;
}
long    c_ymdhms( char *time )
{
        struct  tm  itm             ;
        long        otm             ;
        int         ymdhms          ;
        char        ctm[3]          ;
        ctm[2]      = ‘\0’          ;
        strncpy( ctm , &time[2] , 2 ) ;
        ymdhms      = atoi( ctm )  ;
        itm.tm_year = ymdhms       ;
        strncpy( ctm , &time[4] , 2 ) ;
        ymdhms      = atoi( ctm )  ;
        itm.tm_mon  = ymdhms - 1   ;
        strncpy( ctm , &time[6] , 2 ) ;
        ymdhms      = atoi( ctm )  ;
        itm.tm_mday = ymdhms       ;
        strncpy( ctm , &time[8] , 2 ) ;
        ymdhms      = atoi( ctm )  ;
        itm.tm_hour = ymdhms       ;
        strncpy( ctm , &time[10], 2 ) ;
        ymdhms      = atoi( ctm )  ;
        itm.tm_min  = ymdhms       ;
        strncpy( ctm , &time[12], 2 ) ;
        ymdhms      = atoi( ctm )  ;
        itm.tm_sec  = ymdhms       ;
        itm.tm_isdst = -1          ;
        otm = mktime( &itm )       ;
        return( otm )              ;
}
int TPCC(info,num)
TPSVCINFO *info;
int  num;
{
        int    mix;
        int    k;
        char   logname[80];
        FILE *fp;
        int  rtnsize;
#ifdef NT
        DWORD    work,yDay,yDiff,work_day;
#else
long int    work,yDay,yDiff,work_day;
#endif
#ifdefUSE_FML
        mix = Fvall( ( FBFR * )info->data, FML_TRAN, 0 );
#else
        mix = *((int *)info->data);
#endif
#ifdef  TRACE
        DP(“TPCC-call mix=%d \n”,mix);
#endif
        if( mix == 1 )
        {
#ifdefUSE_FML
            nbuf = *( ( neworder_trans * )Ffind( ( FBFR * )info->data,
```

```
                FML_DATA, 0, NULL ) );                              bpo = &obuf;
        bpn = &nbuf;                              #else
#else                                                     bpo = (orderstat_trans*)info->data;
        bpn = (neworder_trans *)info->data;       #endif
#endif                                                    rtnsize = sizeof(orderstat_trans);
        rtnsize = sizeof(neworder_trans);                 w_id = bpo->w_id ;
        w_id = bpn->w_id ;                                d_id = bpo->d_id ;
        d_id = bpn->d_id ;                                c_id = bpo->c_id;
        bpn->C_R = 0;                                     bpo->C_R = 0;
        tmp_d_id = bpn->d_id;                             c_w_id = bpo->w_id;
        c_id = bpn->c_id;                                 c_d_id =bpo->d_id;
        o_ol_cnt = bpn->o_ol_cnt;                         strncpy(c_last," ",sizeof(c_last));
        bpn->C_R = NewOrder();                            strcpy(c_last,bpo->c_last);
#ifdef TAMESHI_OZA                                        for(k=0 ; k<16; k++)
                if(bpn->C_R==2){                          {
                        FILE *fpoza;                          if (c_last[k] == 0x00)
                        printf("C_R==2\n");                           {
                        fpoza=fopen("/tmp/svrout.oza","a+");                  c_last[k] = 0x20;
                        fprintf(fpoza,"C_R==2 p=%d \n",getpid());            }
                        fflush(fpoza);                           }
                }                                         if(OrderStatus())
#endif                                                    {         bpo->C_R = 1; }
#ifdefUSE_FML                                             else
        Fchg( ( FBFR * )info->data, FML_DATA, 0, ( char * )bpn,   {         bpo->C_R = 0; }
            sizeof( neworder_trans ) );           #ifdefUSE_FML
#endif                                                    Fchg( ( FBFR * )info->data, FML_DATA, 0, ( char * )bpo,
    }                                                         sizeof( orderstat_trans ) );
    else if( mix == 2 )                           #endif
    {                                                 }
#ifdefUSE_FML                                         else if( mix == 4 )
        pbuf = *( ( payment_trans * )Ffind( ( FBFR * )info->data,   {
            FML_DATA, 0, NULL ) );                #ifdefUSE_FML
        bpp = &pbuf;                                      dbuf = *( ( delivery_trans * )Ffind( ( FBFR * )info->data,
#else                                                         FML_DATA, 0, NULL ) );
        bpp = (payment_trans *)info->data;                bpd = &dbuf;
#endif                                             #else
        rtnsize = sizeof(payment_trans);                  bpd = (delivery_trans*)info->data;
        w_id = bpp->w_id ;                        #endif
        d_id = bpp->d_id ;                                w_id = bpd->w_id ;
        c_d_id = bpp->c_d_id;                             d_id = bpd->d_id ;
        c_w_id = bpp->c_w_id;                             bpd->C_R = 0;
        strncpy(c_last," ",sizeof(c_last));               if ( delivery_handle == NULL )
        strcpy(c_last,bpp->c_last);                       {
        for(k=0 ; k<16; k++)                      #ifdefUSE_FML
        {                                                     num = getpid();
            if (c_last[k] == 0x00)                #endif
                    {                                         sprintf(logname,"/tpcrt/tpcc/delivery_log%d",num);
                            c_last[k] = 0x20;                 delivery_handle = fopen(logname,"w+");
                    }                                         if ( delivery_handle == NULL )
        }                                                     {         delivery_handle = stderr;
        c_id = bpp->c_id;                         #ifdef  TRACE
        h_amount = bpp->h_amount;                                 DP("delivery_log1 cannot write\n");
        bpp->C_R = Payment();                     #endif
#ifdefUSE_FML                                                      printf("delivery_log cannot write\n");
        Fchg( ( FBFR * )info->data, FML_DATA, 0, ( char * )bpp,            fflush(stdout);
            sizeof( payment_trans ) );                        }
#endif                                                    }
    }                                                 o_carrier_id = bpd->o_carrier_id;
    else if( mix == 3 )                               for(d_id = 0; d_id < 10; d_id++)
    {                                                 {     result_o_id[d_id ] = 0;        }
#ifdefUSE_FML                                             bpd->C_R = Delivery();
        obuf = *( ( orderstat_trans * )Ffind( ( FBFR * )info->data,   Gettimeofday(&tp_e);
            FML_DATA, 0, NULL ) );                #ifdef  TRACE
```

```
        DP("A-deli_handle= %x C_R=%d \n",delivery_handle,bpd-
>C_R);
#endif
        if(bpd->C_R)
        {
#ifdef NT
        fprintf(delivery_handle,"%09d%03d %09d%03d %d %d",
                bpd->startsec,
                bpd->startusec,
                tp_e.tv_sec,
                tp_e.tv_usec,
                w_id,
                o_carrier_id);
#else
        fprintf(delivery_handle,"%09d%03d %09d%03d %d %d",
                bpd->startsec,
                bpd->startusec/1000,
                tp_e.tv_sec,
                tp_e.tv_usec/1000,
                w_id,
                o_carrier_id);
#endif
        for(d_id = 0; d_id < 10; d_id++)
        {
            fprintf(delivery_handle," %d
%d",d_id+1,result_o_id[d_id]);
        }
        fprintf(delivery_handle, "\n");
        }
        else
        {
#ifdef NT
        fprintf(delivery_handle,"%09d%03d %09d%03d %d %d",
                bpd->startsec,
                bpd->startusec,
                0,
                0,
                w_id,
                o_carrier_id);
#else
        fprintf(delivery_handle,"%09d%03d %09d%03d %d %d",
                bpd->startsec,
                bpd->startusec/1000,
                0,
                0,
                w_id,
                o_carrier_id);
#endif
        fprintf(delivery_handle," errpos:%04d SQLSTATE:%05d\n",
                    bpd->errorpos,bpd->sqlstate);
        }
#ifdef FFLUSH_OUT
        fflush(delivery_handle);
#endif
#ifdef  TRACE
    DP("Out-deli_handle= %x C_R=%d \n",delivery_handle,bpd-
>C_R);
#endif
#ifdef  TRACE
   DP("tpreturn-called MIX =%d TPNOREPLY \n",mix);
#endif
#ifdefUSE_FML
        tpreturn( TPSUCCESS, 0, (char *)NULL, 0, 0 );
```

```
#else
    tpreturn(TPSUCCESS,0,(char *)bpd,sizeof(delivery_trans),0        );
#endif
    }
    else if( mix == 5 )
    {
#ifdefUSE_FML
        sbuf = *( ( stocklvl_trans * )Ffind( ( FBFR * )info->data,
            FML_DATA, 0, NULL ) );
        bps = &sbuf;
#else
    bps = (stocklvl_trans *)info->data;
#endif
    rtnsize = sizeof(stocklvl_trans);
    w_id = bps->w_id ;
    d_id = bps->d_id ;
    threshold = bps->threshold;
    if(StockLevel())
    {
        bps->C_R = 1;
        bps->low_stock = low_stock;
    }
     else
    {
        bps->C_R = 0;
    }
#ifdefUSE_FML
    Fchg( ( FBFR * )info->data, FML_DATA, 0, ( char * )bps,
        sizeof( stocklvl_trans ) );
#endif
    }
    if( mix != 4 )
    {
#ifdef  TRACE
   DP("tpreturn-called mix=%d \n",mix);
#endif
#ifdefUSE_FML
    tpreturn( TPSUCCESS, 0, info->data, 0L, 0 );
#else
    tpreturn(TPSUCCESS,0,info->data,rtnsize,0);
#endif
    }
}
void tpsvrdone()
{
#ifdef  TRACE
    DP("tpsvrdone called pid=%d\n",getpid());
#endif
    JMPCINT3()              ;
    fflush(delivery_handle);
    fclose(delivery_handle);
#ifdefUSE_SQL_MODE
    EXEC SQL COMMIT WORK   ;
#endif
    EXEC SQL DISCONNECT CURRENT ;
    DP("DISCONNECT(SQLSTATE) = %s\n", SQLSTATE) ;
    return;
}
int Error()
{
    char msg[1024];
    long errno;
    FILE *handle;
```

```
        SQLSTATE[5] = 0 ;
        if (0 != strcmp(SQLSTATE,"00000") )
          {
            if (0 == strcmp(SQLSTATE,"40001"))
              {
                return(1);
              }
#ifdef NT
                    system("date /T>>tpccerr");
#else
            system("date >>/tmp/tpccerr");
#endif
#ifdef NT
        handle = fopen("tpccerr","ab");
#else
        handle = fopen("/tmp/tpccerr","ab");
#endif
        if ( handle == NULL )
          {
            handle = stderr;
          }
        fprintf(handle," SQL ERROR:SQLSTATE= %s\n",SQLSTATE);
        fflush(handle);
        }
        return(0);
}
tpsvrinit(argc,argv)
int argc;
char **argv;
{
        int i = 0;
        char *fname;
#ifdef suzuki
        if(tpsvrinit_fp == 0){
                sprintf(fname,"tpsvrinit_test%d.txt",getpid());
                tpsvrinit_fp = fopen(fname,"w");
        }
        fprintf(tpsvrinit_fp,"tpsvrinit start\n");
        fflush(tpsvrinit_fp);
#endif
        DP("tpsvrinit start  called pid=%d\n",getpid());
#ifdefUSE_SQL_MODE
        EXEC SQL WHENEVER SQLERROR CONTINUE;
#endif
#ifdefUSE_SQL_MODE
            EXEC SQL CONNECT TO 'SV1';
#endif
            DP("CONNECT(SQLSTATE) = %s\n", SQLSTATE) ;
#ifdefUSE_SQL_MODE
        EXEC SQL COMMIT WORK ;
#endif
        JMPCINT2();
        if(! preNewOrder() ) ++i;
        if(! prePayment() ) ++i;
        if(! preOrderStatus() ) ++i;
        if(! preDelivery() ) ++i;
        if(! preStockLevel() ) ++i;
        if(i)
        {
                printf("%d errors in SQL prepares.quitting.\n",i);
                fflush(stdout);
                exit(1);
        }
```

```
        DP("tpsvrinit end  called pid=%d\n",getpid());
#ifdef suzuki
        if(tpsvrinit_fp == 0){
                tpsvrinit_fp = fopen("tpsvrinit_test.txt","w");
        }
        fprintf(tpsvrinit_fp,"tpsvrinit end\n");
        fflush(tpsvrinit_fp);
#endif
}
int scanstring(target,search,length)
char *target,*search;
int length;
{
        int search_length,iter;
        if((search_length = strlen(search)) > length)
        {
                return(-1);
        }
        for (iter= length -search_length;iter;--iter,++target)
        {
                if(strncmp(target,search,search_length) == 0)
                {
                        return(1);
                }
        }
        return(0);
}
preNewOrder()
{
    return(1);
}
NewOrder()
{
    long i_price[15];
    char i_name[15][25];
    char i_data[15][51];
    char s_datax[15][51];
      lnk_ol lnk_buf[15];
    int j ;
    int i ;
    long total_amount = 0;
    int  pos = 0;
    int in_ol_i_id          ;
    int in_ol_number         ;
        s_join_str    *sjp   ;
        i_join_str    *ijp   ;
        ol_i_join_str *olijp  ;
        ol_s_join_str *olsjp  ;
        ol_q_join_str *olqjp  ;
    int  item_notfound_cnt     ;
    FILE *fp;
    FILE *handle;
    int  retry_flag = 0 ;
short    *ol_i_id_ind_str[]  = { (short *)&ol_i_id1_ind   ,
                                 (short *)&ol_i_id2_ind   ,
                                 (short *)&ol_i_id3_ind   ,
                                 (short *)&ol_i_id4_ind   ,
                                 (short *)&ol_i_id5_ind   ,
                                 (short *)&ol_i_id6_ind   ,
                                 (short *)&ol_i_id7_ind   ,
                                 (short *)&ol_i_id8_ind   ,
                                 (short *)&ol_i_id9_ind   ,
                                 (short *)&ol_i_id10_ind   ,
```

```
                     (short *)&ol_i_id11_ind   ,                              }
                     (short *)&ol_i_id12_ind   ,                          else
                     (short *)&ol_i_id13_ind   ,                          {
                     (short *)&ol_i_id14_ind   ,                              r_id[r_cnt].num       = ol_number             ;
                     (short *)&ol_i_id15_ind   ,                              r_id[r_cnt].ol_i_id  = bpn->ol_i_id[ol_number]  ;
                     NULL};                                                   r_cnt = r_cnt + 1                            ;
  struct {                                                                 }
        int   num      ;                                              }
        long  ol_i_id    ;                                        }
     } sort_id[15]      ;                                retry_neworder:
  struct {                                                   if ( ( h_cnt < 5 ) || ( retry_flag == 1 ) )
        int   num      ;                                        {
        long  ol_i_id    ;                                        h_cnt = 0 ;
     } r_id[15]          ;                                         r_cnt = 0 ;
        int   sort_num    ;                                        for (ol_number = 0; ol_number < o_ol_cnt ;++ol_number)
        long  sort_ol_i_id ;                                       {
#ifdefUSE_SQL_MODE                                                     r_id[r_cnt].num       = ol_number             ;
EXEC SQL BEGIN DECLARE SECTION;                                        r_id[r_cnt].ol_i_id  = bpn->ol_i_id[ol_number]  ;
#endif                                                                 r_cnt = r_cnt + 1                            ;
        short  h_cnt      ;                                            }
        short  r_cnt      ;                                        }
#ifdefUSE_SQL_MODE                                              for ( i=0 ; i < r_cnt ; i++ )
EXEC SQL END   DECLARE SECTION;                                 {
#endif                                                                  sort_id[h_cnt+i].num    = r_id[i].num          ;
begin_tran:                                                             sort_id[h_cnt+i].ol_i_id = r_id[i].ol_i_id       ;
#ifdefUSE_SQL_MODE                                              }
   EXEC SQL WHENEVER SQLERROR  GOTO :sqlerr    ;                for (ol_number = 0; ol_number < h_cnt ;++ol_number)
   EXEC SQL WHENEVER NOT FOUND GOTO :not_found ;                {
#endif                                                            for (in_ol_number = ol_number + 1  ;
        errorpos        = 0                   ;                       in_ol_number < h_cnt          ;++in_ol_number)
        item_notfound    = -1                    ;                  {
        s_join.sqllen   = 0                    ;                      if (sort_id[in_ol_number].ol_i_id > sort_id[ol_number].ol_i_id)
        i_join.sqllen    = 0                    ;                        {
        sjp          = (s_join_str *)&s_join       ;                    sort_num                = sort_id[ol_number].num      ;
        ijp          = (i_join_str *)&i_join       ;                    sort_ol_i_id             = sort_id[ol_number].ol_i_id  ;
        olijp         = (ol_i_join_str *)&ol_i_join ;                   sort_id[ol_number].num     = sort_id[in_ol_number].num    ;
        olsjp         = (ol_s_join_str *)&ol_s_join ;                   sort_id[ol_number].ol_i_id   = sort_id[in_ol_number].ol_i_id;
        olqjp         = (ol_q_join_str *)&ol_q_join ;                   sort_id[in_ol_number].num     = sort_num               ;
        h_cnt = 0 ;                                                     sort_id[in_ol_number].ol_i_id = sort_ol_i_id             ;
        r_cnt = 0 ;                                                   }
        for (ol_number = 0; ol_number < o_ol_cnt ;++ol_number)      }
        {                                                         }
          if ( w_id == bpn->ol_supply_w_id[ol_number] )        for (i=0,ol_number = 0; ol_number < 15 ;++ol_number)
           {                                                    {
            for ( i=0 ; i < h_cnt ; i++ )                        if (ol_number < h_cnt)
              {                                                    {
               if ( sort_id[i].ol_i_id == bpn->ol_i_id[ol_number] )    *((short *)(ol_i_id_ind_str[ol_number])) = 0         ;
                 {                                                     *((long  *)(ol_i_id_str[ol_number]))
                   break ;                                                 = bpn->ol_i_id[sort_id[ol_number].num]         ;
                 }                                                     sprintf(olqjp->sqlvar[ol_number].ol_quantity,"%-4d",
              }                                                           bpn->ol_quantity[sort_id[ol_number].num] )     ;
            if ( i == h_cnt )                                         }
              {                                                     else
               sort_id[h_cnt].num     = ol_number             ;      {
               sort_id[h_cnt].ol_i_id = bpn->ol_i_id[ol_number]  ;     *((short *)(ol_i_id_ind_str[ol_number])) = -1         ;
               h_cnt = h_cnt + 1                            ;          *((long  *)(ol_i_id_str[ol_number]))      = 0         ;
              }                                                       if ( ol_number < o_ol_cnt )
            else                                                        {
              {                                                          sprintf(olqjp->sqlvar[ol_number].ol_quantity,"%-4d",
               r_id[r_cnt].num       = ol_number             ;             bpn->ol_quantity[sort_id[ol_number].num] )    ;
               r_id[r_cnt].ol_i_id  = bpn->ol_i_id[ol_number]  ;           sprintf(olijp->sqlvar[i].ol_i_id,"%-7d",
               r_cnt = r_cnt + 1                            ;               bpn->ol_i_id[sort_id[ol_number].num] )       ;
              }                                                          sprintf(olsjp->sqlvar[i].ol_supply_w_id,"%-4d",
                                                                           bpn->ol_supply_w_id[sort_id[ol_number].num] ) ;
```

```
                i++ ;
            }
        }
    }
    ol_q_join.sqllen = o_ol_cnt * 4            ;
    ol_i_join.sqllen = r_cnt    * 7            ;
    ol_s_join.sqllen = r_cnt    * 4            ;
    s_ymdhms()                               ;
    strncpy(o_entry_d, tc_s,14)              ;
    bpn->o_entry_d = t_wk                    ;
neworder_proc:
#ifdefUSE_SQL_MODE
 EXEC SQL WHENEVER SQLERROR CONTINUE ;
 EXEC SQL WHENEVER NOT FOUND CONTINUE;
        EXEC SQL
            CALL TPCC_SCHEMA.Y_NORDER5(:state            ,
                :errorpos          INDICATOR :errorpos_ind    ,
                :w_id                                ,
                :tmp_d_id                            ,
                :c_id                                ,
                :o_all_local       INDICATOR :o_all_local_ind  ,
                :w_tax             INDICATOR :w_tax_ind      ,
                :d_tax             INDICATOR :d_tax_ind      ,
                :o_id              INDICATOR :o_id_ind       ,
                :o_entry_d                           ,
                :c_discount        INDICATOR :c_discount_ind   ,
                :c_last            INDICATOR :c_last_ind      ,
                :c_credit          INDICATOR :c_credit_ind     ,
                :item_notfound     INDICATOR :item_notfound_ind ,
                :h_cnt                               ,
                :r_cnt                               ,
                 :ol_i_id1                           ,
                 :ol_i_id2                           ,
                 :ol_i_id3                           ,
                 :ol_i_id4                           ,
                 :ol_i_id5                           ,
                 :ol_i_id6         INDICATOR :ol_i_id6_ind     ,
                 :ol_i_id7         INDICATOR :ol_i_id7_ind     ,
                 :ol_i_id8         INDICATOR :ol_i_id8_ind     ,
                 :ol_i_id9         INDICATOR :ol_i_id9_ind     ,
                 :ol_i_id10        INDICATOR :ol_i_id10_ind    ,
                 :ol_i_id11        INDICATOR :ol_i_id11_ind    ,
                 :ol_i_id12        INDICATOR :ol_i_id12_ind    ,
                 :ol_i_id13        INDICATOR :ol_i_id13_ind    ,
                 :ol_i_id14        INDICATOR :ol_i_id14_ind    ,
                 :ol_i_id15        INDICATOR :ol_i_id15_ind    ,
                :ol_i_join                           ,
                :ol_q_join                           ,
                :s_join                              ,
                :i_join                              ,
                :ol_s_join
                            );
    EXEC SQL WHENEVER SQLERROR  GOTO :sqlerr   ;
#ifdef  TAMESHI_OZA
{
        FILE *fpoza;
            fpoza=fopen("/tmp/svrout.oza","a+");
        fprintf(fpoza,"%d endNOstored  state(%s) SQLST=(%s) p=%d
\n",
                      item_notfound,state,SQLSTATE,getpid());
        fflush(fpoza);
                    fclose(fpoza);
    }
#endif
#else
        SQLWAIT_N;
    strcpy(state,"00000");
    o_id = 3001;
    bpn->w_tax = rand()%2001;
    bpn->d_tax = rand()%2001;
    strcpy( bpn->c_last, "BAROUGHTABLE" );
    strcpy( bpn->c_credit, "GC" );
    bpn->c_discount = rand()%101;
#endif
    if( SQLSTATE[0]== '4' || SQLSTATE[0] == '7' ){
            goto  sqlerr;
    }
    if ( memcmp(state,"00000",5) != 0 )
     {
       strncpy(SQLSTATE,state,5) ;
       SQLSTATE[5] = 0          ;
       if ( memcmp(state,"02000",5) == 0 )
        {
          goto not_found;
        }
       else
        {
          goto sqlerr;
        }
     }
    if ( item_notfound == -2 )
     {
        FILE *fpoza;
            retry_flag = 1;
        item_notfound = -1   ;
        i_join.sqllen = 0  ;
#ifdef  TAMESHI_OZA
        printf("retry  st  \n");
        fpoza=fopen("/tmp/svrout.oza","a+");
        fprintf(fpoza,"retry st   p=%d \n",getpid());
        fflush(fpoza);
                    fclose(fpoza);
#endif
       goto retry_neworder ;
     }
#ifdef  TAMESHI_OZA
     {
        FILE *fpoza;
            fpoza=fopen("/tmp/svrout.oza","a+");
        fprintf(fpoza,"%d  SQLST=(%s) p=%d \n",
                    item_notfound,SQLSTATE,getpid());
        fflush(fpoza);
                    fclose(fpoza);
     }
#endif
#ifdef  DP_IJ
            DP("IJ: item_notfound=%d \n", item_notfound);
#endif
    bpn->o_id    = o_id        ;
#ifdef TAMESHI_OZA
        if( retry_flag != 0 ){
            o_ol_cnt_kari  =  o_ol_cnt - 1;
        }else{
            o_ol_cnt_kari  =  o_ol_cnt;
        }
#endif
```

```
      for ( ol_number = 0;ol_number < o_ol_cnt;++ol_number )                               = bpn->ol_supply_w_id[ol_number] ;
      {                                                                    lnk_buf[ol_number].ol_quantity
        ol_i_id = bpn->ol_i_id[ol_number] ;                                             = bpn->ol_quantity[ol_number]    ;
        for ( in_ol_number = 0;in_ol_number < o_ol_cnt;++in_ol_number      lnk_buf[ol_number].ol_amount      = ol_amount          ;
)                                                                          strncpy(lnk_buf[ol_number].ol_dist_info,ol_dist_info,25) ;
        {                                                                  }
          if ( ol_number == sort_id[in_ol_number].num )           #ifdefUSE_SQL_MODE
            {                                                          errorpos = 108 ;
              i_price[ol_number]                                       j = OLINSERT(&lnk_buf[0],&o_ol_cnt,&SQLSTATE);
                    = atoi(ijp->sqlvar[in_ol_number].i_price)   ;   #ifdef  TAMESHI_OZA
              if ( i_price[ol_number] == 0 )                                   {
                {                                                             FILE *fpoza;
                  bpn->i_price[ol_number]    = 0           ;                  fpoza=fopen("/tmp/svrout.oza","a+");
                  bpn->s_quantity[ol_number] = 0           ;                  fprintf(fpoza,"%d %d=olins   SQLST=(%s) p=%d \n",
                  ol_dist_info[0]            = '\0'        ;                      item_notfound,j,SQLSTATE,getpid());
                  bpn->i_name[ol_number][0]  = '\0'        ;                  fflush(fpoza);
                  break ;                                                     fclose(fpoza);
                }                                                          }
              else                                                 #endif
                {                                                      if ( j != 0)
                  bpn->i_price[ol_number] = i_price[ol_number]      ;    {
                  strncpy(bpn->i_name[ol_number],                           DP(" NewOrder ERRPOS=%d SQLSTATE=%s\n",errorpos, SQLSTATE );
                          ijp->sqlvar[in_ol_number].i_name,24)    ;        goto sqlerr;
                  bpn->i_name[ol_number][24] = '\0'            ;         }
                  strncpy(i_data[ol_number],                             if ( item_notfound == -1)
                          ijp->sqlvar[in_ol_number].i_data,50)    ;      {
                  i_data[ol_number][50]  = '\0'              ;             EXEC SQL COMMIT WORK                ;
                  bpn->s_quantity[ol_number]                               strncpy(bpn->c_last,c_last,17)           ;
                      = atoi(sjp->sqlvar[in_ol_number].s_quantity) ;       strncpy(bpn->c_credit,c_credit,3)        ;
                  strncpy(ol_dist_info,                                    bpn->d_tax      = d_tax             ;
                          sjp->sqlvar[in_ol_number].s_dist,24)    ;        bpn->w_tax      = w_tax             ;
                  ol_dist_info[24] = '\0'                    ;             bpn->c_discount = c_discount            ;
                  strncpy(s_datax[ol_number],                              total_amount *= (1 + (w_tax + d_tax)/10000.0)
                          sjp->sqlvar[in_ol_number].s_data,50)    ;                * (1 - (c_discount    /10000.0)) ;
                  s_datax[ol_number][50]       = '\0'            ;         bpn->total_amount = total_amount          ;
                  break ;                                                  bpn->errorpos    = 0                  ;
                }                                                         bpn->sqlstate    = 0                  ;
      #ifdef  DP_IJ                                                        return(1)                         ;
              DP("IJ: ol_num=%d,", ol_number);                         }
              DP("   price =%d,", i_price[ol_number]);               else
              DP("   name =%s \n", &(bpn->i_name[ol_number][0]) );     {
      #endif                                                              strncpy(bpn->c_last,c_last,17)           ;
            }                                                             strncpy(bpn->c_credit,c_credit,3)        ;
          }                                                               bpn->errorpos    = 201               ;
        ol_amount           = bpn->ol_quantity[ol_number]                 bpn->sqlstate    = 02000             ;
                        * i_price[ol_number]        ;                     EXEC SQL ROLLBACK WORK                 ;
        bpn->ol_amount[ol_number] = ol_amount              ;              return(2)                         ;
        total_amount           += ol_amount            ;               }
        if ( scanstring(i_data[ol_number],"ORIGINAL",50)        #else
          && scanstring(s_datax[ol_number],"ORIGINAL",50) )            SQLWAIT_N_C;
          {                                                             SQLWAIT_N_R;
            bpn->brand_generic[ol_number] = 'B';                    bpn->total_amount = 0;
          }                                                         for ( i = 0; i < 15; ++i )
        else                                                        {
          {                                                           if ( bpn->ol_supply_w_id[i] == 0 ) {
            bpn->brand_generic[ol_number] = 'G';                        break;
          }                                                           }
        lnk_buf[ol_number].ol_o_id      = o_id             ;          strcpy( bpn->i_name[i], "NAMESNAMESNAMESNAME" );
        lnk_buf[ol_number].ol_d_id      = tmp_d_id         ;          bpn->s_quantity[i] = ( rand()%10 ) + 1;
        lnk_buf[ol_number].ol_w_id      = w_id             ;          bpn->brand_generic[i] = 'G';
        lnk_buf[ol_number].ol_number    = ol_number + 1    ;          bpn->i_price[i] = ( rand()%9901 )+100;
        lnk_buf[ol_number].ol_i_id      = ol_i_id          ;          bpn->ol_amount[i] = bpn->i_price[i]*bpn->ol_quantity[i];
        lnk_buf[ol_number].ol_supply_w_id                             bpn->total_amount += bpn->ol_amount[i];
```

```
            }
        bpn->o_ol_cnt = i;                                        :c_id                              ,
                return(1)                           ;            :c_d_id                            ,
#endif                                                           :c_w_id                            ,
not_found:                                                       :h_amount                          ,
    DP("NOT FOUND IN NewOrder AT %d\n",errorpos);               :h_date                            ,
    fflush(stdout);                                              :w_name                            ,
    bpn->errorpos    = errorpos            ;                    :w_street_1       INDICATOR :w_street_1_ind    ,
    bpn->sqlstate    = atoi(SQLSTATE)        ;                  :w_street_2       INDICATOR :w_street_2_ind    ,
#ifdefUSE_SQL_MODE                                               :w_city           INDICATOR :w_city_ind       ,
    EXEC SQL ROLLBACK WORK;                                      :w_state          INDICATOR :w_state_ind      ,
#else                                                            :w_zip            INDICATOR :w_zip_ind        ,
        SQLWAIT_N_R;                                             :d_name                            ,
#endif                                                           :d_street_1       INDICATOR :d_street_1_ind   ,
    return(0);                                                   :d_street_2       INDICATOR :d_street_2_ind   ,
sqlerr:                                                          :d_city           INDICATOR :d_city_ind       ,
#ifdef  DP_SQLERR                                                :d_state          INDICATOR :d_state_ind      ,
    DP("Neworder ERRPOS=%d SQLSTATE=%s\n",errorpos,SQLSTATE);   :d_zip            INDICATOR :d_zip_ind        ,
#endif                                                           :c_first          INDICATOR :c_first_ind      ,
#ifdefUSE_SQL_MODE                                               :c_middle         INDICATOR :c_middle_ind     ,
    EXEC SQL WHENEVER SQLERROR  CONTINUE ;                       :c_last                            ,
#endif                                                           :c_street_1       INDICATOR :c_street_1_ind   ,
    if(Error())                                                  :c_street_2       INDICATOR :c_street_2_ind   ,
        {                                                        :c_city           INDICATOR :c_city_ind       ,
#ifdefUSE_SQL_MODE                                               :c_state          INDICATOR :c_state_ind      ,
        EXEC SQL ROLLBACK WORK;                                  :c_zip            INDICATOR :c_zip_ind        ,
#else                                                            :c_phone          INDICATOR :c_phone_ind      ,
        SQLWAIT_N_R;                                             :c_credit                          ,
#endif                                                           :c_credit_lim     INDICATOR :c_credit_lim_ind ,
            goto begin_tran;                                     :c_discount       INDICATOR :c_discount_ind   ,
        }                                                        :c_balance        INDICATOR :c_balance_ind    ,
    bpn->errorpos    = errorpos            ;                    :c_ytd_payment    INDICATOR :c_ytd_payment_ind ,
    bpn->sqlstate    = atoi(SQLSTATE)        ;                  :c_payment_cnt    INDICATOR :c_payment_cnt_ind ,
#ifdefUSE_SQL_MODE                                               :c_since          INDICATOR :c_since_ind ,
    EXEC SQL ROLLBACK WORK;                                      :c_datax          INDICATOR :c_data_ind
#else                                                                          );
        SQLWAIT_N_R;                                    #else
#endif                                                          SQLWAIT_P;
    return(0);                                                  strcpy(state,"00000");
}                                                               c_discount = rand()%5001;
prePayment()                                                    strcpy( c_first, "ABCDEFGHIJKL" );
{                                                               strcpy( c_middle, "OE" );
    return(1);                                                  strcpy( c_last, "BAROUGHTABLE" );
}                                                               strcpy( c_phone, "0123456789012345" );
Payment()                                                       c_id = rand()%3000 + 1;
{                                                               strcpy( c_street_1, "PQRSTUVWXYZABCD" );
begin_tran:                                                     strcpy( c_street_2, "EFGHIJKOLMNOPQR" );
#ifdefUSE_SQL_MODE                                              strcpy( c_city, "STUVWXYZABCDEFG" );
    EXEC SQL WHENEVER SQLERROR  GOTO :sqlerr    ;               strcpy( c_state, "RE" );
    EXEC SQL WHENEVER NOT FOUND GOTO :not_found ;               sprintf( c_zip, "%04d11111", rand()%10000 );
#endif                                                          strcpy( d_street_1, "PQRSTUVWXYZABCD" );
    s_ymdhms()                  ;                               strcpy( d_street_2, "EFGHIJKOLMNOPQR" );
    strncpy(h_date, tc_s,14)        ;                           strcpy( d_city, "STUVWXYZABCDEFG" );
    c_datax.sqllen      = 0;                                    strcpy( d_state, "RE" );
    c_datax.sqlvar[0]   = 0;                                    sprintf( d_zip, "%04d11111", rand()%10000 );
    c_datax.sqlvar[500] = 0;                                    strcpy( w_street_1, "PQRSTUVWXYZABCD" );
        errorpos = 0            ;                               strcpy( w_street_2, "EFGHIJKOLMNOPQR" );
#ifdefUSE_SQL_MODE                                              strcpy( w_city, "STUVWXYZABCDEFG" );
        EXEC SQL                                                strcpy( w_state, "RE" );
            CALL TPCC_SCHEMA.Y_PAYMENT_H10_OUT4(:state         sprintf( w_zip, "%04d11111", rand()%10000 );
                :errorpos        INDICATOR :errorpos_ind    ,  c_balance = ( ( rand()*rand()%19999999 )-9999999 ) / 100.0;
                :w_id                            ,             c_credit_lim = 5000000;
                :d_id                            ,             strcpy( c_since, "19980212121212" );
                                                               strcpy( c_credit, "GC" );
```

```
#endif
    if ( memcmp(state,"00000",5) != 0 )
      {
        strncpy(SQLSTATE,state,5) ;
        SQLSTATE[5] = 0          ;
        if ( memcmp(state,"02000",5) == 0 )
          {
            goto not_found;
          }
        else
          {
            goto sqlerr;
          }
      }
  bpp->c_discount = c_discount      ;
  bpp->h_date    = t_wk             ;
  strcpy(bpp->c_first,c_first)      ;
  strcpy(bpp->c_middle,c_middle)    ;
  strcpy(bpp->c_last,c_last)        ;
  strcpy(bpp->c_phone,c_phone)      ;
  bpp->c_id= c_id                   ;
  strcpy(bpp->c_street_1,c_street_1) ;
  strcpy(bpp->c_street_2,c_street_2) ;
  strcpy(bpp->c_city,c_city)        ;
  strcpy(bpp->c_state,c_state)      ;
  strcpy(bpp->c_zip,c_zip)          ;
  strcpy(bpp->d_street_1,d_street_1) ;
  strcpy(bpp->d_street_2,d_street_2) ;
  strcpy(bpp->d_city,d_city)        ;
  strcpy(bpp->d_state,d_state)      ;
  strcpy(bpp->d_zip,d_zip)          ;
  strcpy(bpp->w_street_1,w_street_1) ;
  strcpy(bpp->w_street_2,w_street_2) ;
  strcpy(bpp->w_city,w_city)        ;
  strcpy(bpp->w_state,w_state)      ;
  strcpy(bpp->w_zip,w_zip)          ;
  bpp->c_balance = c_balance ⁄ 100.0 ;
  bpp->c_credit_lim = c_credit_lim ⁄ 100.0 ;
  bpp->c_since = c_ymdhms(c_since)   ;
  strcpy(bpp->c_credit,c_credit)    ;
#ifdefUSE_SQL_MODE
 if ( strcmp(c_credit,"BC") == 0 )
  {
     strncpy(bpp->c_data,c_datax.sqlvar,c_datax.sqllen);
  }
 else
  {
     bpp->c_data[0] = 0 ;
  }
#else
 if ( rand()%10 == 1 ) {
#define _STR50"0123456789abcdefghijklmnopqrstuvwxyz!#$&'()=-
~[]::"
   strcpy( bpp->c_credit, "BC" );
   strcpy( bpp->c_data,
    _STR50 _STR50 _STR50 _STR50 _STR50 _STR50 _STR50 _STR50 );
 } else {
   bpp->c_data[0] = '\0';
 }
#endif
 bpp->errorpos    = 0 ;
 bpp->sqlstate    = 0 ;
 return(1);
```

```
not_found:
    DP("NOT FOUND IN Payment AT %d\n",errorpos);
    fflush(stdout);
    bpp->errorpos     = errorpos           ;
    bpp->sqlstate     = atoi(SQLSTATE)        ;
    return(0);
sqlerr:
#ifdef DP_SQLERR
    DP("Payment  ERRPOS=%d SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
  if(Error())
  {
     goto begin_tran;
   }
  bpp->errorpos     = errorpos           ;
  bpp->sqlstate     = atoi(SQLSTATE)        ;
  return(0);
}
preOrderStatus()
{
  return(1);
}
OrderStatus()
{
    ol_join_str  *oljp                ;
    int l;
    char ol_supply_w_id_5[5] = {0,0,0,0,0};
begin_tran:
#ifdefUSE_SQL_MODE
  EXEC SQL WHENEVER SQLERROR  GOTO :sqlerr    ;
  EXEC SQL WHENEVER NOT FOUND GOTO :not_found ;
#endif
    ol_join.sqllen = 0                  ;
    oljp       = (ol_join_str *)&ol_join ;
    errorpos = 0       ;
#ifdefUSE_SQL_MODE
    EXEC SQL
       CALL TPCC_SCHEMA.Y_ODERSTAT(:state            ,
           :errorpos         INDICATOR :errorpos_ind     ,
           :w_id                          ,
           :d_id                          ,
           :c_id                          ,
           :c_first       INDICATOR :c_first_ind      ,
           :c_middle        INDICATOR :c_middle_ind     ,
           :c_last        INDICATOR :c_last_ind       ,
           :c_balance       INDICATOR :c_balance_ind    ,
           :o_id          INDICATOR :o_id_ind        ,
           :o_entry_d      INDICATOR :o_entry_d_ind     ,
           :o_carrier_id    INDICATOR :o_carrier_id_ind  ,
           :o_ol_cnt                     ,
           :ol_join        INDICATOR :ol_join_ind
                      );
#else
      SQLWAIT_O;
    strcpy(state,"00000");
    c_id = rand()%3000 + 1;
    strcpy( c_first, "ABCDEFGHIJKL" );
    strcpy( c_middle, "OE" );
    strcpy( c_last, "BAROUGHTABLE" );
    c_balance = ( ( rand()*rand()%19999999 )-9999999 ) ⁄ 100.0;
    o_id = rand()%99999999+1;
    strcpy( o_entry_d, "19980123123456" );
    o_ol_cnt = rand()%11 + 5;
```

```
#endif
    if ( memcmp(state,"00000",5) != 0 )
      {
        strncpy(SQLSTATE,state,5) ;
        SQLSTATE[5] = 0           ;
        if ( memcmp(state,"02000",5) == 0 )
          {
            goto not_found;
          }
        else
          {
            goto sqlerr;
          }
      }
#ifdefUSE_SQL_MODE
    for ( ol_number = 0;ol_number < o_ol_cnt;++ol_number )
      {
        bpo->ol_i_id[ol_number]   = atoi(oljp->sqlvar[ol_number].ol_i_id) ;
        bpo->ol_amount[ol_number] = atoi(oljp-
>sqlvar[ol_number].ol_amount);
                for(l = 0;l < 4;l++){
                    ol_supply_w_id_5[l] = oljp-
>sqlvar[ol_number].ol_supply_w_id[l];
                }
        bpo->ol_supply_w_id[ol_number]
                = atoi(ol_supply_w_id_5)   ;
        bpo->ol_quantity[ol_number]
                    = atoi(oljp->sqlvar[ol_number].ol_quantity)      ;
        if(memcmp(oljp->sqlvar[ol_number].ol_delivery_d,"777777777",9) !=
0)
          {
            bpo->ol_delivery_d[ol_number]
                    = c_ymdhms(oljp->sqlvar[ol_number].ol_delivery_d);
          }
        else
          {
            bpo->ol_delivery_d[ol_number] = 777777777              ;
          }
      }
#else
    for ( ol_number = 0; ol_number < o_ol_cnt; ++ol_number )
      {
        bpo->ol_i_id[ol_number] = ( rand()%100000 )+1;
        bpo->ol_amount[ol_number] = rand()%1000000;
        bpo->ol_supply_w_id[ol_number] = ( rand()%10 )+1;
        bpo->ol_quantity[ol_number] = ( rand()%99 )+1;
        bpo->ol_delivery_d[ol_number] = c_ymdhms( "19980321054321" );
      }
#endif
    if ( o_carrier_id_ind == -1 )
      {
        bpo->o_carrier_id = INTNULL        ;
      }
    else
      {
        bpo->o_carrier_id = o_carrier_id   ;
      }
    bpo->c_id        = c_id           ;
    bpo->o_ol_cnt    = o_ol_cnt       ;
    strcpy(bpo->c_first,c_first)      ;
    strcpy(bpo->c_middle,c_middle)     ;
    strcpy(bpo->c_last,c_last)         ;
    bpo->c_balance   = c_balance/100.0 ;


        bpo->o_id        = o_id             ;
        bpo->o_entry_d   = c_ymdhms(o_entry_d) ;
        bpo->errorpos    = 0               ;
        bpo->sqlstate    = 0               ;
    return (1);
not_found:
    DP("NOT FOUND IN OrderStatus AT %d\n",errorpos);
    fflush(stdout);
    bpo->errorpos    = errorpos            ;
    bpo->sqlstate    = atoi(SQLSTATE)       ;
    return(0);
sqlerr:
#ifdef  DP_SQLERR
    DP("OrderStatus ERRPOS=%d SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
    if(Error())
      {
        goto begin_tran;
      }
    bpo->errorpos    = errorpos            ;
    bpo->sqlstate    = atoi(SQLSTATE)       ;
    return(0);
}
preDelivery()
{
    return(1);
}
Delivery()
{
        int     temp_d_id                 ;
            result_join_str *rjp           ;
begin_tran:
#ifdefUSE_SQL_MODE
        EXEC SQL WHENEVER SQLERROR  GOTO :sqlerr    ;
        EXEC SQL WHENEVER NOT FOUND GOTO :not_found ;
#endif
        result_join.sqllen = 0                 ;
        rjp = (result_join_str *)&result_join    ;
        s_ymdhms()                    ;
        strncpy(ol_delivery_d, tc_s,14)          ;
        o_carrier_id  = bpd->o_carrier_id       ;
        errorpos      = 0                 ;
#ifdefUSE_SQL_MODE
        EXEC SQL
            CALL TPCC_SCHEMA.Y_DELIVERY(:state                  ,
                :errorpos         INDICATOR :errorpos_ind    ,
                :w_id                           ,
                :c_id                           ,
                :o_carrier_id                   ,
                :ol_delivery_d                  ,
                :result_join       INDICATOR :result_join_ind
                    );
#else
        SQLWAIT_D;
    strcpy(state,"00000");
#endif
    if ( memcmp(state,"00000",5) != 0 )
      {
        strncpy(SQLSTATE,state,5) ;
        SQLSTATE[5] = 0           ;
        if ( memcmp(state,"02000",5) == 0 )
          {
            goto not_found;
```

```
            }
          else
            {
              goto sqlerr;
            }
        }
      for ( temp_d_id = 0 ; temp_d_id < 10 ; temp_d_id++ )
        {
           result_o_id[temp_d_id] = atoi(rjp-
>sqlvar[temp_d_id].result_o_id);
        }
      bpd->errorpos    = 0 ;
      bpd->sqlstate    = 0 ;
      return(1);
not_found:
      DP("NOT FOUND IN DELIVERY AT %d\n",errorpos);
      fflush(stdout);
      bpd->errorpos    = errorpos          ;
      bpd->sqlstate    = atoi(SQLSTATE)        ;
      return(0);
sqlerr:
#ifdef  DP_SQLERR
   DP("Delivery ERRPOS=%d SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
      if(Error())
      {
            goto begin_tran;
      }
      bpd->errorpos    = errorpos          ;
      bpd->sqlstate    = atoi(SQLSTATE)        ;
      return(0);
}
preStockLevel()
{
      return(1);

}
StockLevel()
{
begin_tran:
#ifdefUSE_SQL_MODE
      EXEC SQL WHENEVER SQLERROR  GOTO :sqlerr    ;
      EXEC SQL WHENEVER NOT FOUND GOTO :not_found ;
#endif
        errorpos = 0                      ;
#ifdef STOCK_STORED
#ifdefUSE_SQL_MODE
        EXEC SQL
          CALL TPCC_SCHEMA.Y_STOCKLV(:state          ,
                :errorpos         INDICATOR :errorpos_ind ,
                :w_id                             ,
                :d_id                             ,
                :threshold                        ,
                :low_stock        INDICATOR :low_stock_ind
                             );
#else
      SQLWAIT_S;
      strcpy(state,"00000");
      low_stock = rand()%201;
#endif
      if ( memcmp(state,"00000",5) != 0 )
        {
          strncpy(SQLSTATE,state,5) ;
          SQLSTATE[5] = 0            ;
```

```
      if ( memcmp(state,"02000",5) == 0 )
        {
          goto not_found;
        }
      else
        {
          goto sqlerr;
        }
    }
#else
              EXEC SQL WHENEVER SQLERROR  GOTO :ERR_S_DI;
              EXEC SQL WHENEVER NOT FOUND GOTO :ERR_S_DI;
      EXEC SQL SELECT D_NEXT_O_ID
          INTO :o_id
          FROM TPCC_SCHEMA.DISTRICT
          WHERE D_W_ID = :w_id
            AND D_ID   = :d_id;
              EXEC SQL WHENEVER SQLERROR  CONTINUE;
              EXEC SQL WHENEVER NOT FOUND CONTINUE;
      tmp_o_id = o_id - 20;
      o_id = o_id - 1 ;
      t19  = o_id - 1;
      t18  = o_id - 2;
      t17  = o_id - 3;
      t16  = o_id - 4;
      t15  = o_id - 5;
      t14  = o_id - 6;
      t13  = o_id - 7;
      t12  = o_id - 8;
      t11  = o_id - 9;
      t10  = o_id - 10;
      t09  = o_id - 11;
      t08  = o_id - 12;
      t07  = o_id - 13;
      t06  = o_id - 14;
      t05  = o_id - 15;
      t04  = o_id - 16;
      t03  = o_id - 17;
      t02  = o_id - 18;
                EXEC SQL WHENEVER SQLERROR  GOTO :ERR_S_STOL;
                EXEC SQL WHENEVER NOT FOUND GOTO :ERR_S_STOL;
      EXEC SQL SELECT COUNT(DISTINCT S_I_ID)
          INTO  :low_stock
          FROM  TPCC_SCHEMA.ORDERLINE,
              TPCC_SCHEMA.STOCK
          WHERE OL_W_ID   = :w_id
           AND OL_D_ID    = :d_id
           AND OL_O_ID
             IN(:tmp_o_id,
                  :t02,:t03,:t04,:t05,:t06,:t07,:t08,:t09,:t10,
                :t11,:t12,:t13,:t14,:t15,:t16,:t17,:t18,:t19,
                :o_id )
           AND OL_NUMBER IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
--$
--$       AND OL_O_ID
--$               BETWEEN @TMP_O_ID
--$                  AND @O_ID
           AND S_I_ID    = OL_I_ID
           AND S_W_ID    = OL_W_ID
           AND S_W_ID    = :w_id
           AND S_QUANTITY < :threshold;
                EXEC SQL WHENEVER SQLERROR  CONTINUE;
                EXEC SQL WHENEVER NOT FOUND CONTINUE;
```

```
        EXEC SQL COMMIT WORK;                                    number = 5;
#endif                                                           return(TPCC(info,number));
    bps->errorpos    = 0 ;                                       }
    bps->sqlstate    = 0 ;                               TPCC6(info)
    return(1);                                           TPSVCINFO      *info;
ERR_S_DI:                                                        {
    bps->errorpos  = 203;                                        number = 6;
        goto sqlerr;                                             return(TPCC(info,number));
ERR_S_STOL:                                                      }
    bps->errorpos  = 248;                                TPCC7(info)
        goto sqlerr;                                     TPSVCINFO      *info;
not_found:                                                       {
    DP("NOT FOUND IN STOCKLEVEL AT %d\n",errorpos);              number = 7;
    fflush(stdout);                                              return(TPCC(info,number));
    bps->errorpos    = errorpos         ;                        }
    bps->sqlstate    = atoi(SQLSTATE)   ;                TPCC8(info)
    EXEC SQL ROLLBACK WORK;                              TPSVCINFO      *info;
    return(0);                                                   {
sqlerr:                                                          number = 8;
#ifdef  DP_SQLERR                                                return(TPCC(info,number));
   DP("Stocklevel ERRPOS=%d SQLSTATE=%s\n",errorpos,SQLSTATE);   }
#endif                                                   TPCC9(info)
    if(Error())                                          TPSVCINFO      *info;
    {                                                            {
        EXEC SQL ROLLBACK WORK;                                  number = 9;
        goto begin_tran;                                         return(TPCC(info,number));
    }                                                            }
    bps->errorpos    = errorpos         ;                TPCC10(info)
    bps->sqlstate    = atoi(SQLSTATE)   ;                TPSVCINFO      *info;
    EXEC SQL ROLLBACK WORK;                                      {
    return(0);                                                   number = 10;
}                                                                return(TPCC(info,number));
#ifndefUSE_FML                                                   }
TPCC1(info)                                              TPCCd1(info)
TPSVCINFO      *info;                                    TPSVCINFO      *info;
    {                                                            {
#ifdef  TRACE                                                    number = 11;
    DP("TPCC-1 called\n");                                       return(TPCC(info,number));
#endif                                                           }
    number = 1;                                          TPCCd2(info)
    return(TPCC(info,number));                           TPSVCINFO      *info;
    }                                                            {
TPCC2(info)                                                      number = 12;
TPSVCINFO      *info;                                            return(TPCC(info,number));
    {                                                            }
    number = 2;                                          TPCCd3(info)
    return(TPCC(info,number));                           TPSVCINFO      *info;
    }                                                            {
TPCC3(info)                                                      number = 13;
TPSVCINFO      *info;                                            return(TPCC(info,number));
    {                                                            }
    number = 3;                                          TPCCs1(info)
    return(TPCC(info,number));                           TPSVCINFO      *info;
    }                                                            {
TPCC4(info)                                                      number = 14;
TPSVCINFO      *info;                                            return(TPCC(info,number));
    {                                                            }
    number = 4;                                          TPCCs2(info)
    return(TPCC(info,number));                           TPSVCINFO      *info;
    }                                                            {
TPCC5(info)                                                      number = 15;
TPSVCINFO      *info;                                            return(TPCC(info,number));
    {                                                            }
```

A

```
TPCCs3(info)
TPSVCINFO        *info;
        {
        number = 16;
        return(TPCC(info,number));
        }
#endif
```

This Appendix contains the scripts used to create the database and the load program used to load the database initially.

```
::::::::::::::
::::::::::::::
ALL.mk
::::::::::::::
:
set -x

rdbstop

CRDIC.notlog

date

rdbstart

rdbddlex ddl_db.mak
rdbddlex -d TPCC crta.def.cent

rdbddlex -d TPCC ddl.dbsp.192.dat

cd 2112wh-for-E450-CU4k

rdbddlex -d TPCC ddl.dat.WH
rdbddlex -d TPCC ddl.dat.DI
rdbddlex -d TPCC ddl.dat.ST
rdbddlex -d TPCC ddl.dat.NO
rdbddlex -d TPCC ddl.dat.NI
rdbddlex -d TPCC ddl.dat.OS
rdbddlex -d TPCC ddl.dat.OI
rdbddlex -d TPCC ddl.dat.HI
rdbddlex -d TPCC ddl.dat.CU.4k
rdbddlex -d TPCC ddl.dat.CI
rdbddlex -d TPCC ddl.dat.OL
rdbddlex -d TPCC ddl.dat.IT
```

```
cd ..

sleep 5
rdbstop
date

rdbstart
sh.stored
sleep 5
rdbstop

rdbstart
cd stored/Y_stored.PS
sh Y_stored_cent.sh.h10.out4
cd ../..
rdbstop


rdbstart
timex csh -x LOAD.sh

rdbstop

rdbstart

sh.rdbups
rdbstop

sleep 5
::::::::::::::
LOAD.sh
::::::::::::::
#!/bin/csh -xf

setenv RDBDB TPCC

set LOAD1_D = /rdb/loaddata/2
set LOAD2_D = /rdb/loaddata/3
set LOAD3_D = /rdb/loaddata/4
set LOAD4_D = /rdb/loaddata/6
```

```
set WK1_D = /rdb/sortwk1
set WK2_D = /rdb/sortwk2
set WK3_D = /rdb/sortwk3
set WK4_D = /rdb/sortwk4
set WK5_D = /rdb/sortwk5
set WK6_D = /rdb/sortwk6
set WK7_D = /rdb/sortwk7
set WK8_D = /rdb/sortwk8

rm /rdb/loaddata/*/??*_* /rdb/loaddata/*/data
rm /rdb/sortwk*/SRT*

## Item

    wttpccd1 $LOAD1_D 1 1 I
    timex rdbsloader -mi -i $RDBDB.ITEM_1_DSI \
                                    -s $WK1_D \
                                    -s $WK2_D \
                                    -n $LOAD1_D/data
rm $LOAD1_D/data

## WAREHOUSE

foreach num ( 0 1 2 3 4 5 6 )

    @ make_s1 = $num * 352 + 1
    @ make_e1 = $num * 352 + 88
    @ make_s2 = $num * 352 + 89
    @ make_e2 = $num * 352 + 176
    @ make_s3 = $num * 352 + 177
    @ make_e3 = $num * 352 + 264
    @ make_s4 = $num * 352 + 265
    @ make_e4 = $num * 352 + 352

    @ dsi_num1 = $num * 4 + 1
    @ dsi_num2 = $num * 4 + 2
    @ dsi_num3 = $num * 4 + 3
    @ dsi_num4 = $num * 4 + 4
    wttpccd1 $LOAD1_D $make_s1 $make_e1 W &
    wttpccd1 $LOAD2_D $make_s2 $make_e2 W &
    wttpccd1 $LOAD3_D $make_s3 $make_e3 W &
    wttpccd1 $LOAD4_D $make_s4 $make_e4 W &

    wait

if ( $num != 0 ) then

    @ rm_make_s1 = $num * 352 + 1 - 352
    @ rm_make_e1 = $num * 352 + 88 - 352
    @ rm_make_s2 = $num * 352 + 89 - 352
    @ rm_make_e2 = $num * 352 + 176 - 352
    @ rm_make_s3 = $num * 352 + 177 - 352
    @ rm_make_e3 = $num * 352 + 264 - 352
    @ rm_make_s4 = $num * 352 + 265 - 352
    @ rm_make_e4 = $num * 352 + 352 - 352
    rm $LOAD1_D/WH$rm_make_s1\_$rm_make_e1
    rm $LOAD2_D/WH$rm_make_s2\_$rm_make_e2
    rm $LOAD3_D/WH$rm_make_s3\_$rm_make_e3
    rm $LOAD4_D/WH$rm_make_s4\_$rm_make_e4
endif

if ( $dsi_num1 <= 25 ) then
```

```
    timex rdbsloader -mi -i $RDBDB.WAREHOUSE_$dsi_num1\_DSI \
                                    -s $WK1_D \
                                    -s $WK2_D \
                                    -n
$LOAD1_D/WH$make_s1\_$make_e1 &
endif
if ( $dsi_num2 <= 25 ) then
    timex rdbsloader -mi -i $RDBDB.WAREHOUSE_$dsi_num2\_DSI \
                                    -s $WK3_D \
                                    -s $WK4_D \
                                    -n
$LOAD2_D/WH$make_s2\_$make_e2 &
endif
if ( $dsi_num3 <= 25 ) then
    timex rdbsloader -mi -i $RDBDB.WAREHOUSE_$dsi_num3\_DSI \
                                    -s $WK5_D \
                                    -s $WK6_D \
                                    -n
$LOAD3_D/WH$make_s3\_$make_e3 &
endif
if ( $dsi_num4 <= 25 ) then
    timex rdbsloader -mi -i $RDBDB.WAREHOUSE_$dsi_num4\_DSI \
                                    -s $WK7_D \
                                    -s $WK8_D \
                                    -n
$LOAD4_D/WH$make_s4\_$make_e4 &
endif
end

    wait
rm /rdb/loaddata/*/WH*
## DISTRICT

foreach num ( 0 1 2 3 4 5 6 )

    @ make_s1 = $num * 352 + 1
    @ make_e1 = $num * 352 + 88
    @ make_s2 = $num * 352 + 89
    @ make_e2 = $num * 352 + 176
    @ make_s3 = $num * 352 + 177
    @ make_e3 = $num * 352 + 264
    @ make_s4 = $num * 352 + 265
    @ make_e4 = $num * 352 + 352

    @ dsi_num1 = $num * 4 + 1
    @ dsi_num2 = $num * 4 + 2
    @ dsi_num3 = $num * 4 + 3
    @ dsi_num4 = $num * 4 + 4
    wttpccd1 $LOAD1_D $make_s1 $make_e1 D &
    wttpccd1 $LOAD2_D $make_s2 $make_e2 D &
    wttpccd1 $LOAD3_D $make_s3 $make_e3 D &
    wttpccd1 $LOAD4_D $make_s4 $make_e4 D &

    wait

if ( $num != 0 ) then

    @ rm_make_s1 = $num * 352 + 1 - 352
    @ rm_make_e1 = $num * 352 + 88 - 352
    @ rm_make_s2 = $num * 352 + 89 - 352
    @ rm_make_e2 = $num * 352 + 176 - 352
    @ rm_make_s3 = $num * 352 + 177 - 352
    @ rm_make_e3 = $num * 352 + 264 - 352
```

```
        @ rm_make_s4 = $num * 352 + 265 - 352
        @ rm_make_e4 = $num * 352 + 352 - 352
        rm $LOAD1_D/DI$rm_make_s1\_$rm_make_e1
        rm $LOAD2_D/DI$rm_make_s2\_$rm_make_e2
        rm $LOAD3_D/DI$rm_make_s3\_$rm_make_e3
        rm $LOAD4_D/DI$rm_make_s4\_$rm_make_e4
endif

if ( $dsi_num1 <= 25 ) then
        timex rdbsloader -mi -i $RDBDB.DISTRICT_$dsi_num1\_DSI \
                                        -s $WK1_D \
                                        -s $WK2_D \
                                        -n
$LOAD1_D/DI$make_s1\_$make_e1 &
endif
if ( $dsi_num2 <= 25 ) then
        timex rdbsloader -mi -i $RDBDB.DISTRICT_$dsi_num2\_DSI \
                                        -s $WK3_D \
                                        -s $WK4_D \
                                        -n
$LOAD2_D/DI$make_s2\_$make_e2 &
endif
if ( $dsi_num3 <= 25 ) then
        timex rdbsloader -mi -i $RDBDB.DISTRICT_$dsi_num3\_DSI \
                                        -s $WK5_D \
                                        -s $WK6_D \
                                        -n
$LOAD3_D/DI$make_s3\_$make_e3 &
endif
if ( $dsi_num4 <= 25 ) then
        timex rdbsloader -mi -i $RDBDB.DISTRICT_$dsi_num4\_DSI \
                                        -s $WK7_D \
                                        -s $WK8_D \
                                        -n
$LOAD4_D/DI$make_s4\_$make_e4 &
endif
end

        wait
rm /rdb/loaddata/*/DI*
## CUSTOMER

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 )

        @ make_s1 = $num * 88 + 1
        @ make_e1 = $num * 88 + 22
        @ make_s2 = $num * 88 + 23
        @ make_e2 = $num * 88 + 44
        @ make_s3 = $num * 88 + 45
        @ make_e3 = $num * 88 + 66
        @ make_s4 = $num * 88 + 67
        @ make_e4 = $num * 88 + 88

        @ dsi_num1 = $num * 4 + 1
        @ dsi_num2 = $num * 4 + 2
        @ dsi_num3 = $num * 4 + 3
        @ dsi_num4 = $num * 4 + 4
        wttpccd1 $LOAD1_D $make_s1 $make_e1 C &
        wttpccd1 $LOAD2_D $make_s2 $make_e2 C &
        wttpccd1 $LOAD3_D $make_s3 $make_e3 C &
        wttpccd1 $LOAD4_D $make_s4 $make_e4 C &
```

```
        wait

if ( $num != 0 ) then

        @ rm_make_s1 = $num * 88 + 1 - 88
        @ rm_make_e1 = $num * 88 + 22 - 88
        @ rm_make_s2 = $num * 88 + 23 - 88
        @ rm_make_e2 = $num * 88 + 44 - 88
        @ rm_make_s3 = $num * 88 + 45 - 88
        @ rm_make_e3 = $num * 88 + 66 - 88
        @ rm_make_s4 = $num * 88 + 67 - 88
        @ rm_make_e4 = $num * 88 + 88 - 88
        rm $LOAD1_D/CU$rm_make_s1\_$rm_make_e1
        rm $LOAD2_D/CU$rm_make_s2\_$rm_make_e2
        rm $LOAD3_D/CU$rm_make_s3\_$rm_make_e3
        rm $LOAD4_D/CU$rm_make_s4\_$rm_make_e4
endif

if ( $dsi_num1 <= 100 ) then
        timex rdbsloader -mi -i $RDBDB.CUSTOMER_$dsi_num1\_DSI -h \
                                        -s $WK1_D \
                                        -s $WK2_D \
                                        -n
$LOAD1_D/CU$make_s1\_$make_e1 &
endif
if ( $dsi_num2 <= 100 ) then
        timex rdbsloader -mi -i $RDBDB.CUSTOMER_$dsi_num2\_DSI -h \
                                        -s $WK3_D \
                                        -s $WK4_D \
                                        -n
$LOAD2_D/CU$make_s2\_$make_e2 &
endif
if ( $dsi_num3 <= 100 ) then
        timex rdbsloader -mi -i $RDBDB.CUSTOMER_$dsi_num3\_DSI -h \
                                        -s $WK5_D \
                                        -s $WK6_D \
                                        -n
$LOAD3_D/CU$make_s3\_$make_e3 &
endif
if ( $dsi_num4 <= 100 ) then
        timex rdbsloader -mi -i $RDBDB.CUSTOMER_$dsi_num4\_DSI -h \
                                        -s $WK7_D \
                                        -s $WK8_D \
                                        -n
$LOAD4_D/CU$make_s4\_$make_e4 &
endif
end

        wait
rm /rdb/loaddata/*/CU*
## HISTORY

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 )

        @ make_s1 = $num * 88 + 1
        @ make_e1 = $num * 88 + 22
        @ make_s2 = $num * 88 + 23
        @ make_e2 = $num * 88 + 44
        @ make_s3 = $num * 88 + 45
        @ make_e3 = $num * 88 + 66
        @ make_s4 = $num * 88 + 67
        @ make_e4 = $num * 88 + 88
```

```
    @ dsi_num1 = $num * 4 + 1
    @ dsi_num2 = $num * 4 + 2
    @ dsi_num3 = $num * 4 + 3
    @ dsi_num4 = $num * 4 + 4
    wttpccd1 $LOAD1_D $make_s1 $make_e1 H &
    wttpccd1 $LOAD2_D $make_s2 $make_e2 H &
    wttpccd1 $LOAD3_D $make_s3 $make_e3 H &
    wttpccd1 $LOAD4_D $make_s4 $make_e4 H &

    wait

if ( $num != 0 ) then

    @ rm_make_s1 = $num * 88 + 1 - 88
    @ rm_make_e1 = $num * 88 + 22 - 88
    @ rm_make_s2 = $num * 88 + 23 - 88
    @ rm_make_e2 = $num * 88 + 44 - 88
    @ rm_make_s3 = $num * 88 + 45 - 88
    @ rm_make_e3 = $num * 88 + 66 - 88
    @ rm_make_s4 = $num * 88 + 67 - 88
    @ rm_make_e4 = $num * 88 + 88 - 88
    rm $LOAD1_D/HI$rm_make_s1\_$rm_make_e1
    rm $LOAD2_D/HI$rm_make_s2\_$rm_make_e2
    rm $LOAD3_D/HI$rm_make_s3\_$rm_make_e3
    rm $LOAD4_D/HI$rm_make_s4\_$rm_make_e4
endif

if ( $dsi_num1 <= 100 ) then
    timex rdbsloader -mi -i $RDBDB.HISTORY_$dsi_num1\_DSI \
                                          -s $WK1_D \
                                          -s $WK2_D \
                                          -n
$LOAD1_D/HI$make_s1\_$make_e1 &
endif
if ( $dsi_num2 <= 100 ) then
    timex rdbsloader -mi -i $RDBDB.HISTORY_$dsi_num2\_DSI \
                                          -s $WK3_D \
                                          -s $WK4_D \
                                          -n
$LOAD2_D/HI$make_s2\_$make_e2 &
endif
if ( $dsi_num3 <= 100 ) then
    timex rdbsloader -mi -i $RDBDB.HISTORY_$dsi_num3\_DSI \
                                          -s $WK5_D \
                                          -s $WK6_D \
                                          -n
$LOAD3_D/HI$make_s3\_$make_e3 &
endif
if ( $dsi_num4 <= 100 ) then
    timex rdbsloader -mi -i $RDBDB.HISTORY_$dsi_num4\_DSI \
                                          -s $WK7_D \
                                          -s $WK8_D \
                                          -n
$LOAD4_D/HI$make_s4\_$make_e4 &
endif
end

    wait
rm /rdb/loaddata/*/HI*

## ORDERS ORDERLINE NEWORDER
```

```
foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 )

    @ make_s1 = $num * 44 + 1
    @ make_e1 = $num * 44 + 11
    @ make_s2 = $num * 44 + 12
    @ make_e2 = $num * 44 + 22
    @ make_s3 = $num * 44 + 23
    @ make_e3 = $num * 44 + 33
    @ make_s4 = $num * 44 + 34
    @ make_e4 = $num * 44 + 44

    @ dsi_num_os1 = $num * 2 + 1
    @ dsi_num_os2 = $num * 2 + 2
    @ dsi_num_no1 = $num * 2 + 1
    @ dsi_num_no2 = $num * 2 + 2
    @ dsi_num_ol1 = $num * 4 + 1
    @ dsi_num_ol2 = $num * 4 + 2
    @ dsi_num_ol3 = $num * 4 + 3
    @ dsi_num_ol4 = $num * 4 + 4
    wttpccd1 $LOAD1_D $make_s1 $make_e1 O &
    wttpccd1 $LOAD2_D $make_s2 $make_e2 O &
    wttpccd1 $LOAD3_D $make_s3 $make_e3 O &
    wttpccd1 $LOAD4_D $make_s4 $make_e4 O &

    wait

if ( $num != 0 ) then

    @ rm_make_s1 = $num * 44 + 1 - 44
    @ rm_make_e1 = $num * 44 + 11 - 44
    @ rm_make_s2 = $num * 44 + 12 - 44
    @ rm_make_e2 = $num * 44 + 22 - 44
    @ rm_make_s3 = $num * 44 + 23 - 44
    @ rm_make_e3 = $num * 44 + 33 - 44
    @ rm_make_s4 = $num * 44 + 34 - 44
    @ rm_make_e4 = $num * 44 + 44 - 44
    rm $LOAD1_D/OS$rm_make_s1\_$rm_make_e1
    rm $LOAD1_D/OL$rm_make_s1\_$rm_make_e1
    rm $LOAD1_D/NO$rm_make_s1\_$rm_make_e1
    rm $LOAD2_D/OS$rm_make_s2\_$rm_make_e2
    rm $LOAD2_D/OL$rm_make_s2\_$rm_make_e2
    rm $LOAD2_D/NO$rm_make_s2\_$rm_make_e2
    rm $LOAD3_D/OS$rm_make_s3\_$rm_make_e3
    rm $LOAD3_D/OL$rm_make_s3\_$rm_make_e3
    rm $LOAD3_D/NO$rm_make_s3\_$rm_make_e3
    rm $LOAD4_D/OS$rm_make_s4\_$rm_make_e4
    rm $LOAD4_D/OL$rm_make_s4\_$rm_make_e4
    rm $LOAD4_D/NO$rm_make_s4\_$rm_make_e4
endif
if ( $dsi_num_os1 <= 100 ) then
    timex rdbsloader -mi -i $RDBDB.ORDERS_$dsi_num_os1\_DSI -h -f 10\
                                          -s $WK1_D \
                                          -s $WK2_D \
                                          $LOAD1_D/OS$make_s1\_$make_e1
\
                                          $LOAD2_D/OS$make_s2\_$make_e2
&
endif
if ( $dsi_num_os2 <= 100 ) then
    timex rdbsloader -mi -i $RDBDB.ORDERS_$dsi_num_os2\_DSI -h -f 10\
                                          -s $WK5_D \
                                          -s $WK6_D \
```

```
$LOAD3_D/OS$make_s3\_$make_e3 \

$LOAD4_D/OS$make_s4\_$make_e4 &
endif

        wait

if ( $dsi_num_no1 <= 100 ) then
     timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no1\_DSI -h -f 20 \
                                  -s $WK1_D \
                                  -s $WK2_D \
                                  -n \

$LOAD1_D/NO$make_s1\_$make_e1 \

$LOAD2_D/NO$make_s2\_$make_e2 &
endif
if ( $dsi_num_no2 <= 100 ) then
     timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no2\_DSI -h -f 20 \
                                  -s $WK5_D \
                                  -s $WK6_D \
                                  -n \

$LOAD3_D/NO$make_s3\_$make_e3 \

$LOAD4_D/NO$make_s4\_$make_e4 &
endif

        wait

if ( $dsi_num_ol1 <= 200 ) then
     timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol1\_DSI -h \
                                  -s $WK1_D \
                                  -s $WK2_D \

$LOAD1_D/OL$make_s1\_$make_e1 &
endif
if ( $dsi_num_ol2 <= 200 ) then
     timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol2\_DSI -h \
                                  -s $WK3_D \
                                  -s $WK4_D \

$LOAD2_D/OL$make_s2\_$make_e2 &
endif
if ( $dsi_num_ol3 <= 200 ) then
     timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol3\_DSI -h \
                                  -s $WK5_D \
                                  -s $WK6_D \

$LOAD3_D/OL$make_s3\_$make_e3 &
endif
if ( $dsi_num_ol4 <= 200 ) then
     timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol4\_DSI -h \
                                  -s $WK7_D \
                                  -s $WK8_D \
```

```
$LOAD4_D/OL$make_s4\_$make_e4 &
endif
end

        wait
rm /rdb/loaddata/*/OS*
rm /rdb/loaddata/*/OL*
rm /rdb/loaddata/*/NO*

LOAD.sh.ST
::::::::::::::
ddl.dat.CI
::::::::::::::
--------------------------------------------------------------------------
-- * Phase.2-3b: Customer-Index
--------------------------------------------------------------------------
     CREATE DSO CUSTOMER_IX_DSO
          INDEX ON TPCC_SCHEMA.CUSTOMER(C_W_ID,C_D_ID,C_LAST)
          TYPE BTREE(PAGESIZE1(16),PAGESIZE2(32));

     CREATE DSI CUSTOMER_X_1DSI
          INDEX
          DSO CUSTOMER_IX_DSO
          BASE CUSTOMER_1_DSI
          ALLOCATE INDEX    ON SP1 SIZE 224K,
                BASE     ON SP1 SIZE 8416K
                                    SP2 SIZE 8400K;

     CREATE DSI CUSTOMER_X_2DSI
          INDEX
          DSO CUSTOMER_IX_DSO
          BASE CUSTOMER_2_DSI
          ALLOCATE INDEX    ON SP3 SIZE 224K,
                BASE     ON SP3 SIZE 8416K
                                    SP4 SIZE 8400K;

     CREATE DSI CUSTOMER_X_3DSI
          INDEX
          DSO CUSTOMER_IX_DSO
          BASE CUSTOMER_3_DSI
          ALLOCATE INDEX    ON SP5 SIZE 224K,
                BASE     ON SP5 SIZE 8416K
                                    SP6 SIZE 8400K;

     CREATE DSI CUSTOMER_X_4DSI
          INDEX
          DSO CUSTOMER_IX_DSO
          BASE CUSTOMER_4_DSI
          ALLOCATE INDEX    ON SP7 SIZE 224K,
                BASE     ON SP7 SIZE 8416K
                                    SP8 SIZE 8400K;

     CREATE DSI CUSTOMER_X_5DSI
          INDEX
          DSO CUSTOMER_IX_DSO
          BASE CUSTOMER_5_DSI
          ALLOCATE INDEX    ON SP9 SIZE 224K,
                BASE     ON SP9 SIZE 8416K
                                    SP10 SIZE 8400K;

     CREATE DSI CUSTOMER_X_6DSI
          INDEX
```

```
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_6_DSI                                    DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP11 SIZE 224K,                  BASE CUSTOMER_14_DSI
                   BASE    ON SP11 SIZE 8416K                     ALLOCATE INDEX    ON SP27 SIZE 224K,
                                     SP12 SIZE 8400K;                    BASE    ON SP27 SIZE 8416K
                                                                                          SP28 SIZE 8400K;
CREATE DSI CUSTOMER_X_7DSI
            INDEX                                          CREATE DSI CUSTOMER_X_15DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_7_DSI                                    DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP13 SIZE 224K,                  BASE CUSTOMER_15_DSI
                   BASE    ON SP13 SIZE 8416K                     ALLOCATE INDEX    ON SP29 SIZE 224K,
                                     SP14 SIZE 8400K;                    BASE    ON SP29 SIZE 8416K
                                                                                          SP30 SIZE 8400K;
CREATE DSI CUSTOMER_X_8DSI
            INDEX                                          CREATE DSI CUSTOMER_X_16DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_8_DSI                                    DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP15 SIZE 224K,                  BASE CUSTOMER_16_DSI
                   BASE    ON SP15 SIZE 8416K                     ALLOCATE INDEX    ON SP31 SIZE 224K,
                                     SP16 SIZE 8400K;                    BASE    ON SP31 SIZE 8416K
                                                                                          SP32 SIZE 8400K;
CREATE DSI CUSTOMER_X_9DSI
            INDEX                                          CREATE DSI CUSTOMER_X_17DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_9_DSI                                    DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP17 SIZE 224K,                  BASE CUSTOMER_17_DSI
                   BASE    ON SP17 SIZE 8416K                     ALLOCATE INDEX    ON SP33 SIZE 224K,
                                     SP18 SIZE 8400K;                    BASE    ON SP33 SIZE 8416K
                                                                                          SP34 SIZE 8400K;
CREATE DSI CUSTOMER_X_10DSI
            INDEX                                          CREATE DSI CUSTOMER_X_18DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_10_DSI                                   DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP19 SIZE 224K,                  BASE CUSTOMER_18_DSI
                   BASE    ON SP19 SIZE 8416K                     ALLOCATE INDEX    ON SP35 SIZE 224K,
                                     SP20 SIZE 8400K;                    BASE    ON SP35 SIZE 8416K
                                                                                          SP36 SIZE 8400K;
CREATE DSI CUSTOMER_X_11DSI
            INDEX                                          CREATE DSI CUSTOMER_X_19DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_11_DSI                                   DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP21 SIZE 224K,                  BASE CUSTOMER_19_DSI
                   BASE    ON SP21 SIZE 8416K                     ALLOCATE INDEX    ON SP37 SIZE 224K,
                                     SP22 SIZE 8400K;                    BASE    ON SP37 SIZE 8416K
                                                                                          SP38 SIZE 8400K;
CREATE DSI CUSTOMER_X_12DSI
            INDEX                                          CREATE DSI CUSTOMER_X_20DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_12_DSI                                   DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP23 SIZE 224K,                  BASE CUSTOMER_20_DSI
                   BASE    ON SP23 SIZE 8416K                     ALLOCATE INDEX    ON SP39 SIZE 224K,
                                     SP24 SIZE 8400K;                    BASE    ON SP39 SIZE 8416K
                                                                                          SP40 SIZE 8400K;
CREATE DSI CUSTOMER_X_13DSI
            INDEX                                          CREATE DSI CUSTOMER_X_21DSI
            DSO CUSTOMER_IX_DSO                                    INDEX
            BASE CUSTOMER_13_DSI                                   DSO CUSTOMER_IX_DSO
            ALLOCATE INDEX    ON SP25 SIZE 224K,                  BASE CUSTOMER_21_DSI
                   BASE    ON SP25 SIZE 8416K                     ALLOCATE INDEX    ON SP41 SIZE 224K,
                                     SP26 SIZE 8400K;                    BASE    ON SP41 SIZE 8416K
                                                                                          SP42 SIZE 8400K;
CREATE DSI CUSTOMER_X_14DSI
```

```
CREATE DSI CUSTOMER_X_22DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_22_DSI
    ALLOCATE INDEX    ON SP43 SIZE 224K,
        BASE    ON SP43 SIZE 8416K
                        SP44 SIZE 8400K;

CREATE DSI CUSTOMER_X_23DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_23_DSI
    ALLOCATE INDEX    ON SP45 SIZE 224K,
        BASE    ON SP45 SIZE 8416K
                        SP46 SIZE 8400K;

CREATE DSI CUSTOMER_X_24DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_24_DSI
    ALLOCATE INDEX    ON SP47 SIZE 224K,
        BASE    ON SP47 SIZE 8416K
                        SP48 SIZE 8400K;

CREATE DSI CUSTOMER_X_25DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_25_DSI
    ALLOCATE INDEX    ON SP49 SIZE 224K,
        BASE    ON SP49 SIZE 8416K
                        SP50 SIZE 8400K;

CREATE DSI CUSTOMER_X_26DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_26_DSI
    ALLOCATE INDEX    ON SP51 SIZE 224K,
        BASE    ON SP51 SIZE 8416K
                        SP52 SIZE 8400K;

CREATE DSI CUSTOMER_X_27DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_27_DSI
    ALLOCATE INDEX    ON SP53 SIZE 224K,
        BASE    ON SP53 SIZE 8416K
                        SP54 SIZE 8400K;

CREATE DSI CUSTOMER_X_28DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_28_DSI
    ALLOCATE INDEX    ON SP55 SIZE 224K,
        BASE    ON SP55 SIZE 8416K
                        SP56 SIZE 8400K;

CREATE DSI CUSTOMER_X_29DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_29_DSI
    ALLOCATE INDEX    ON SP57 SIZE 224K,
        BASE    ON SP57 SIZE 8416K
                        SP58 SIZE 8400K;

CREATE DSI CUSTOMER_X_30DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_30_DSI
    ALLOCATE INDEX    ON SP59 SIZE 224K,
        BASE    ON SP59 SIZE 8416K
                        SP60 SIZE 8400K;

CREATE DSI CUSTOMER_X_31DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_31_DSI
    ALLOCATE INDEX    ON SP61 SIZE 224K,
        BASE    ON SP61 SIZE 8416K
                        SP62 SIZE 8400K;

CREATE DSI CUSTOMER_X_32DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_32_DSI
    ALLOCATE INDEX    ON SP63 SIZE 224K,
        BASE    ON SP63 SIZE 8416K
                        SP64 SIZE 8400K;

CREATE DSI CUSTOMER_X_33DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_33_DSI
    ALLOCATE INDEX    ON SP65 SIZE 224K,
        BASE    ON SP65 SIZE 8416K
                        SP66 SIZE 8400K;

CREATE DSI CUSTOMER_X_34DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_34_DSI
    ALLOCATE INDEX    ON SP67 SIZE 224K,
        BASE    ON SP67 SIZE 8416K
                        SP68 SIZE 8400K;

CREATE DSI CUSTOMER_X_35DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_35_DSI
    ALLOCATE INDEX    ON SP69 SIZE 224K,
        BASE    ON SP69 SIZE 8416K
                        SP70 SIZE 8400K;

CREATE DSI CUSTOMER_X_36DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_36_DSI
    ALLOCATE INDEX    ON SP71 SIZE 224K,
        BASE    ON SP71 SIZE 8416K
                        SP72 SIZE 8400K;

CREATE DSI CUSTOMER_X_37DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_37_DSI
    ALLOCATE INDEX    ON SP73 SIZE 224K,
        BASE    ON SP73 SIZE 8416K
```

```
                              SP74 SIZE 8400K;

CREATE DSI CUSTOMER_X_38DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_38_DSI
     ALLOCATE INDEX    ON SP75 SIZE 224K,
          BASE     ON SP75 SIZE 8416K
                              SP76 SIZE 8400K;

CREATE DSI CUSTOMER_X_39DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_39_DSI
     ALLOCATE INDEX    ON SP77 SIZE 224K,
          BASE     ON SP77 SIZE 8416K
                              SP78 SIZE 8400K;

CREATE DSI CUSTOMER_X_40DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_40_DSI
     ALLOCATE INDEX    ON SP79 SIZE 224K,
          BASE     ON SP79 SIZE 8416K
                              SP80 SIZE 8400K;

CREATE DSI CUSTOMER_X_41DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_41_DSI
     ALLOCATE INDEX    ON SP81 SIZE 224K,
          BASE     ON SP81 SIZE 8416K
                              SP82 SIZE 8400K;

CREATE DSI CUSTOMER_X_42DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_42_DSI
     ALLOCATE INDEX    ON SP83 SIZE 224K,
          BASE     ON SP83 SIZE 8416K
                              SP84 SIZE 8400K;

CREATE DSI CUSTOMER_X_43DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_43_DSI
     ALLOCATE INDEX    ON SP85 SIZE 224K,
          BASE     ON SP85 SIZE 8416K
                              SP86 SIZE 8400K;

CREATE DSI CUSTOMER_X_44DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_44_DSI
     ALLOCATE INDEX    ON SP87 SIZE 224K,
          BASE     ON SP87 SIZE 8416K
                              SP88 SIZE 8400K;

CREATE DSI CUSTOMER_X_45DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_45_DSI
     ALLOCATE INDEX    ON SP89 SIZE 224K,
```

```
          BASE     ON SP89 SIZE 8416K
                              SP90 SIZE 8400K;

CREATE DSI CUSTOMER_X_46DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_46_DSI
     ALLOCATE INDEX    ON SP91 SIZE 224K,
          BASE     ON SP91 SIZE 8416K
                              SP92 SIZE 8400K;

CREATE DSI CUSTOMER_X_47DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_47_DSI
     ALLOCATE INDEX    ON SP93 SIZE 224K,
          BASE     ON SP93 SIZE 8416K
                              SP94 SIZE 8400K;

CREATE DSI CUSTOMER_X_48DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_48_DSI
     ALLOCATE INDEX    ON SP95 SIZE 224K,
          BASE     ON SP95 SIZE 8416K
                              SP96 SIZE 8400K;

CREATE DSI CUSTOMER_X_49DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_49_DSI
     ALLOCATE INDEX    ON SP97 SIZE 224K,
          BASE     ON SP97 SIZE 8416K
                              SP98 SIZE 8400K;

CREATE DSI CUSTOMER_X_50DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_50_DSI
     ALLOCATE INDEX    ON SP99 SIZE 224K,
          BASE     ON SP99 SIZE 8416K
                              SP100 SIZE 8400K;

CREATE DSI CUSTOMER_X_51DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_51_DSI
     ALLOCATE INDEX    ON SP101 SIZE 224K,
          BASE     ON SP101 SIZE 8416K
                              SP102 SIZE 8400K;

CREATE DSI CUSTOMER_X_52DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_52_DSI
     ALLOCATE INDEX    ON SP103 SIZE 224K,
          BASE     ON SP103 SIZE 8416K
                              SP104 SIZE 8400K;

CREATE DSI CUSTOMER_X_53DSI
     INDEX
     DSO CUSTOMER_IX_DSO
     BASE CUSTOMER_53_DSI
```

```
        ALLOCATE INDEX    ON SP105 SIZE 224K,
            BASE     ON SP105 SIZE 8416K
                                SP106 SIZE 8400K;

CREATE DSI CUSTOMER_X_54DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_54_DSI
    ALLOCATE INDEX    ON SP107 SIZE 224K,
        BASE     ON SP107 SIZE 8416K
                            SP108 SIZE 8400K;

CREATE DSI CUSTOMER_X_55DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_55_DSI
    ALLOCATE INDEX    ON SP109 SIZE 224K,
        BASE     ON SP109 SIZE 8416K
                            SP110 SIZE 8400K;

CREATE DSI CUSTOMER_X_56DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_56_DSI
    ALLOCATE INDEX    ON SP111 SIZE 224K,
        BASE     ON SP111 SIZE 8416K
                            SP112 SIZE 8400K;

CREATE DSI CUSTOMER_X_57DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_57_DSI
    ALLOCATE INDEX    ON SP113 SIZE 224K,
        BASE     ON SP113 SIZE 8416K
                            SP114 SIZE 8400K;

CREATE DSI CUSTOMER_X_58DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_58_DSI
    ALLOCATE INDEX    ON SP115 SIZE 224K,
        BASE     ON SP115 SIZE 8416K
                            SP116 SIZE 8400K;

CREATE DSI CUSTOMER_X_59DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_59_DSI
    ALLOCATE INDEX    ON SP117 SIZE 224K,
        BASE     ON SP117 SIZE 8416K
                            SP118 SIZE 8400K;

CREATE DSI CUSTOMER_X_60DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_60_DSI
    ALLOCATE INDEX    ON SP119 SIZE 224K,
        BASE     ON SP119 SIZE 8416K
                            SP120 SIZE 8400K;

CREATE DSI CUSTOMER_X_61DSI
    INDEX
    DSO CUSTOMER_IX_DSO
```

```
        BASE CUSTOMER_61_DSI
        ALLOCATE INDEX    ON SP121 SIZE 224K,
            BASE     ON SP121 SIZE 8416K
                                SP122 SIZE 8400K;

CREATE DSI CUSTOMER_X_62DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_62_DSI
    ALLOCATE INDEX    ON SP123 SIZE 224K,
        BASE     ON SP123 SIZE 8416K
                            SP124 SIZE 8400K;

CREATE DSI CUSTOMER_X_63DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_63_DSI
    ALLOCATE INDEX    ON SP125 SIZE 224K,
        BASE     ON SP125 SIZE 8416K
                            SP126 SIZE 8400K;

CREATE DSI CUSTOMER_X_64DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_64_DSI
    ALLOCATE INDEX    ON SP127 SIZE 224K,
        BASE     ON SP127 SIZE 8416K
                            SP128 SIZE 8400K;

CREATE DSI CUSTOMER_X_65DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_65_DSI
    ALLOCATE INDEX    ON SP129 SIZE 224K,
        BASE     ON SP129 SIZE 8416K
                            SP130 SIZE 8400K;

CREATE DSI CUSTOMER_X_66DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_66_DSI
    ALLOCATE INDEX    ON SP131 SIZE 224K,
        BASE     ON SP131 SIZE 8416K
                            SP132 SIZE 8400K;

CREATE DSI CUSTOMER_X_67DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_67_DSI
    ALLOCATE INDEX    ON SP133 SIZE 224K,
        BASE     ON SP133 SIZE 8416K
                            SP134 SIZE 8400K;

CREATE DSI CUSTOMER_X_68DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_68_DSI
    ALLOCATE INDEX    ON SP135 SIZE 224K,
        BASE     ON SP135 SIZE 8416K
                            SP136 SIZE 8400K;

CREATE DSI CUSTOMER_X_69DSI
    INDEX
```

```
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_69_DSI
        ALLOCATE INDEX    ON SP137 SIZE 224K,
              BASE    ON SP137 SIZE 8416K
                                SP138 SIZE 8400K;

CREATE DSI CUSTOMER_X_70DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_70_DSI
        ALLOCATE INDEX    ON SP139 SIZE 224K,
              BASE    ON SP139 SIZE 8416K
                                SP140 SIZE 8400K;

CREATE DSI CUSTOMER_X_71DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_71_DSI
        ALLOCATE INDEX    ON SP141 SIZE 224K,
              BASE    ON SP141 SIZE 8416K
                                SP142 SIZE 8400K;

CREATE DSI CUSTOMER_X_72DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_72_DSI
        ALLOCATE INDEX    ON SP143 SIZE 224K,
              BASE    ON SP143 SIZE 8416K
                                SP144 SIZE 8400K;

CREATE DSI CUSTOMER_X_73DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_73_DSI
        ALLOCATE INDEX    ON SP145 SIZE 224K,
              BASE    ON SP145 SIZE 8416K
                                SP146 SIZE 8400K;

CREATE DSI CUSTOMER_X_74DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_74_DSI
        ALLOCATE INDEX    ON SP147 SIZE 224K,
              BASE    ON SP147 SIZE 8416K
                                SP148 SIZE 8400K;

CREATE DSI CUSTOMER_X_75DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_75_DSI
        ALLOCATE INDEX    ON SP149 SIZE 224K,
              BASE    ON SP149 SIZE 8416K
                                SP150 SIZE 8400K;

CREATE DSI CUSTOMER_X_76DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_76_DSI
        ALLOCATE INDEX    ON SP151 SIZE 224K,
              BASE    ON SP151 SIZE 8416K
                                SP152 SIZE 8400K;

CREATE DSI CUSTOMER_X_77DSI
```

```
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_77_DSI
        ALLOCATE INDEX    ON SP153 SIZE 224K,
              BASE    ON SP153 SIZE 8416K
                                SP154 SIZE 8400K;

CREATE DSI CUSTOMER_X_78DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_78_DSI
        ALLOCATE INDEX    ON SP155 SIZE 224K,
              BASE    ON SP155 SIZE 8416K
                                SP156 SIZE 8400K;

CREATE DSI CUSTOMER_X_79DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_79_DSI
        ALLOCATE INDEX    ON SP157 SIZE 224K,
              BASE    ON SP157 SIZE 8416K
                                SP158 SIZE 8400K;

CREATE DSI CUSTOMER_X_80DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_80_DSI
        ALLOCATE INDEX    ON SP159 SIZE 224K,
              BASE    ON SP159 SIZE 8416K
                                SP160 SIZE 8400K;

CREATE DSI CUSTOMER_X_81DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_81_DSI
        ALLOCATE INDEX    ON SP161 SIZE 224K,
              BASE    ON SP161 SIZE 8416K
                                SP162 SIZE 8400K;

CREATE DSI CUSTOMER_X_82DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_82_DSI
        ALLOCATE INDEX    ON SP163 SIZE 224K,
              BASE    ON SP163 SIZE 8416K
                                SP164 SIZE 8400K;

CREATE DSI CUSTOMER_X_83DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_83_DSI
        ALLOCATE INDEX    ON SP165 SIZE 224K,
              BASE    ON SP165 SIZE 8416K
                                SP166 SIZE 8400K;

CREATE DSI CUSTOMER_X_84DSI
        INDEX
        DSO CUSTOMER_IX_DSO
        BASE CUSTOMER_84_DSI
        ALLOCATE INDEX    ON SP167 SIZE 224K,
              BASE    ON SP167 SIZE 8416K
                                SP168 SIZE 8400K;
```

```
CREATE DSI CUSTOMER_X_85DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_85_DSI
    ALLOCATE INDEX    ON SP169 SIZE 224K,
        BASE    ON SP169 SIZE 8416K
                        SP170 SIZE 8400K;

CREATE DSI CUSTOMER_X_86DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_86_DSI
    ALLOCATE INDEX    ON SP171 SIZE 224K,
        BASE    ON SP171 SIZE 8416K
                        SP172 SIZE 8400K;

CREATE DSI CUSTOMER_X_87DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_87_DSI
    ALLOCATE INDEX    ON SP173 SIZE 224K,
        BASE    ON SP173 SIZE 8416K
                        SP174 SIZE 8400K;

CREATE DSI CUSTOMER_X_88DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_88_DSI
    ALLOCATE INDEX    ON SP175 SIZE 224K,
        BASE    ON SP175 SIZE 8416K
                        SP176 SIZE 8400K;

CREATE DSI CUSTOMER_X_89DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_89_DSI
    ALLOCATE INDEX    ON SP177 SIZE 224K,
        BASE    ON SP177 SIZE 8416K
                        SP178 SIZE 8400K;

CREATE DSI CUSTOMER_X_90DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_90_DSI
    ALLOCATE INDEX    ON SP179 SIZE 224K,
        BASE    ON SP179 SIZE 8416K
                        SP180 SIZE 8400K;

CREATE DSI CUSTOMER_X_91DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_91_DSI
    ALLOCATE INDEX    ON SP181 SIZE 224K,
        BASE    ON SP181 SIZE 8416K
                        SP182 SIZE 8400K;

CREATE DSI CUSTOMER_X_92DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_92_DSI
    ALLOCATE INDEX    ON SP183 SIZE 224K,
        BASE    ON SP183 SIZE 8416K
                        SP184 SIZE 8400K;

CREATE DSI CUSTOMER_X_93DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_93_DSI
    ALLOCATE INDEX    ON SP185 SIZE 224K,
        BASE    ON SP185 SIZE 8416K
                        SP186 SIZE 8400K;

CREATE DSI CUSTOMER_X_94DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_94_DSI
    ALLOCATE INDEX    ON SP187 SIZE 224K,
        BASE    ON SP187 SIZE 8416K
                        SP188 SIZE 8400K;

CREATE DSI CUSTOMER_X_95DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_95_DSI
    ALLOCATE INDEX    ON SP189 SIZE 224K,
        BASE    ON SP189 SIZE 8416K
                        SP190 SIZE 8400K;

CREATE DSI CUSTOMER_X_96DSI
    INDEX
    DSO CUSTOMER_IX_DSO
    BASE CUSTOMER_96_DSI
    ALLOCATE INDEX    ON SP191 SIZE 224K,
        BASE    ON SP191 SIZE 8416K
                        SP192 SIZE 8400K;

::::::::::::::
ddl.dat.CU.4k
::::::::::::::
--------------------------------------------------------------------------
-- * Phase.2-3a: Customer
--------------------------------------------------------------------------
CREATE DSO CUSTOMER_DSO
    FROM TPCC_SCHEMA.CUSTOMER
        TYPE RANDOM(PAGESIZE1(4),PAGESIZE2(1),
        RULE(C_ID*44+C_W_ID*2+C_D_ID/6+C_D_ID*132000))
        WHERE (C_W_ID) BETWEEN (?) AND (?);

CREATE DSI CUSTOMER_1_DSI
    DSO CUSTOMER_DSO
    USING(1,22)
    ALLOCATE PRIME    ON SP1 SIZE 264004K
                        SP2 SIZE 264000K,
            OVERFLOW ON SP1 SIZE 13201K
                        SP2 SIZE 13200K;

CREATE DSI CUSTOMER_2_DSI
    DSO CUSTOMER_DSO
    USING(23,44)
    ALLOCATE PRIME    ON SP3 SIZE 264004K
                        SP4 SIZE 264000K,
            OVERFLOW ON SP3 SIZE 13201K
                        SP4 SIZE 13200K;

CREATE DSI CUSTOMER_3_DSI
    DSO CUSTOMER_DSO
```

```
        USING(45,66)
        ALLOCATE PRIME    ON SP5 SIZE 264004K
                             SP6 SIZE 264000K,
              OVERFLOW ON SP5 SIZE 13201K
                             SP6 SIZE 13200K;

CREATE DSI CUSTOMER_4_DSI
    DSO CUSTOMER_DSO
    USING(67,88)
    ALLOCATE PRIME    ON SP7 SIZE 264004K
                         SP8 SIZE 264000K,
          OVERFLOW ON SP7 SIZE 13201K
                         SP8 SIZE 13200K;

CREATE DSI CUSTOMER_5_DSI
    DSO CUSTOMER_DSO
    USING(89,110)
    ALLOCATE PRIME    ON SP9 SIZE 264004K
                         SP10 SIZE 264000K,
          OVERFLOW ON SP9 SIZE 13201K
                         SP10 SIZE 13200K;

CREATE DSI CUSTOMER_6_DSI
    DSO CUSTOMER_DSO
    USING(111,132)
    ALLOCATE PRIME    ON SP11 SIZE 264004K
                         SP12 SIZE 264000K,
          OVERFLOW ON SP11 SIZE 13201K
                         SP12 SIZE 13200K;

CREATE DSI CUSTOMER_7_DSI
    DSO CUSTOMER_DSO
    USING(133,154)
    ALLOCATE PRIME    ON SP13 SIZE 264004K
                         SP14 SIZE 264000K,
          OVERFLOW ON SP13 SIZE 13201K
                         SP14 SIZE 13200K;

CREATE DSI CUSTOMER_8_DSI
    DSO CUSTOMER_DSO
    USING(155,176)
    ALLOCATE PRIME    ON SP15 SIZE 264004K
                         SP16 SIZE 264000K,
          OVERFLOW ON SP15 SIZE 13201K
                         SP16 SIZE 13200K;

CREATE DSI CUSTOMER_9_DSI
    DSO CUSTOMER_DSO
    USING(177,198)
    ALLOCATE PRIME    ON SP17 SIZE 264004K
                         SP18 SIZE 264000K,
          OVERFLOW ON SP17 SIZE 13201K
                         SP18 SIZE 13200K;

CREATE DSI CUSTOMER_10_DSI
    DSO CUSTOMER_DSO
    USING(199,220)
    ALLOCATE PRIME    ON SP19 SIZE 264004K
                         SP20 SIZE 264000K,
          OVERFLOW ON SP19 SIZE 13201K
                         SP20 SIZE 13200K;

CREATE DSI CUSTOMER_11_DSI
    DSO CUSTOMER_DSO
    USING(221,242)
    ALLOCATE PRIME    ON SP21 SIZE 264004K
                         SP22 SIZE 264000K,
          OVERFLOW ON SP21 SIZE 13201K
                         SP22 SIZE 13200K;

CREATE DSI CUSTOMER_12_DSI
    DSO CUSTOMER_DSO
    USING(243,264)
    ALLOCATE PRIME    ON SP23 SIZE 264004K
                         SP24 SIZE 264000K,
          OVERFLOW ON SP23 SIZE 13201K
                         SP24 SIZE 13200K;

CREATE DSI CUSTOMER_13_DSI
    DSO CUSTOMER_DSO
    USING(265,286)
    ALLOCATE PRIME    ON SP25 SIZE 264004K
                         SP26 SIZE 264000K,
          OVERFLOW ON SP25 SIZE 13201K
                         SP26 SIZE 13200K;

CREATE DSI CUSTOMER_14_DSI
    DSO CUSTOMER_DSO
    USING(287,308)
    ALLOCATE PRIME    ON SP27 SIZE 264004K
                         SP28 SIZE 264000K,
          OVERFLOW ON SP27 SIZE 13201K
                         SP28 SIZE 13200K;

CREATE DSI CUSTOMER_15_DSI
    DSO CUSTOMER_DSO
    USING(309,330)
    ALLOCATE PRIME    ON SP29 SIZE 264004K
                         SP30 SIZE 264000K,
          OVERFLOW ON SP29 SIZE 13201K
                         SP30 SIZE 13200K;

CREATE DSI CUSTOMER_16_DSI
    DSO CUSTOMER_DSO
    USING(331,352)
    ALLOCATE PRIME    ON SP31 SIZE 264004K
                         SP32 SIZE 264000K,
          OVERFLOW ON SP31 SIZE 13201K
                         SP32 SIZE 13200K;

CREATE DSI CUSTOMER_17_DSI
    DSO CUSTOMER_DSO
    USING(353,374)
    ALLOCATE PRIME    ON SP33 SIZE 264004K
                         SP34 SIZE 264000K,
          OVERFLOW ON SP33 SIZE 13201K
                         SP34 SIZE 13200K;

CREATE DSI CUSTOMER_18_DSI
    DSO CUSTOMER_DSO
    USING(375,396)
    ALLOCATE PRIME    ON SP35 SIZE 264004K
                         SP36 SIZE 264000K,
          OVERFLOW ON SP35 SIZE 13201K
                         SP36 SIZE 13200K;
```

```
CREATE DSI CUSTOMER_19_DSI
    DSO CUSTOMER_DSO
     USING(397,418)
     ALLOCATE PRIME    ON SP37 SIZE 264004K
                          SP38 SIZE 264000K,
          OVERFLOW ON SP37 SIZE 13201K
                          SP38 SIZE 13200K;

CREATE DSI CUSTOMER_20_DSI
    DSO CUSTOMER_DSO
     USING(419,440)
     ALLOCATE PRIME    ON SP39 SIZE 264004K
                          SP40 SIZE 264000K,
          OVERFLOW ON SP39 SIZE 13201K
                          SP40 SIZE 13200K;

CREATE DSI CUSTOMER_21_DSI
    DSO CUSTOMER_DSO
     USING(441,462)
     ALLOCATE PRIME    ON SP41 SIZE 264004K
                          SP42 SIZE 264000K,
          OVERFLOW ON SP41 SIZE 13201K
                          SP42 SIZE 13200K;

CREATE DSI CUSTOMER_22_DSI
    DSO CUSTOMER_DSO
     USING(463,484)
     ALLOCATE PRIME    ON SP43 SIZE 264004K
                          SP44 SIZE 264000K,
          OVERFLOW ON SP43 SIZE 13201K
                          SP44 SIZE 13200K;

CREATE DSI CUSTOMER_23_DSI
    DSO CUSTOMER_DSO
     USING(485,506)
     ALLOCATE PRIME    ON SP45 SIZE 264004K
                          SP46 SIZE 264000K,
          OVERFLOW ON SP45 SIZE 13201K
                          SP46 SIZE 13200K;

CREATE DSI CUSTOMER_24_DSI
    DSO CUSTOMER_DSO
     USING(507,528)
     ALLOCATE PRIME    ON SP47 SIZE 264004K
                          SP48 SIZE 264000K,
          OVERFLOW ON SP47 SIZE 13201K
                          SP48 SIZE 13200K;

CREATE DSI CUSTOMER_25_DSI
    DSO CUSTOMER_DSO
     USING(529,550)
     ALLOCATE PRIME    ON SP49 SIZE 264004K
                          SP50 SIZE 264000K,
          OVERFLOW ON SP49 SIZE 13201K
                          SP50 SIZE 13200K;

CREATE DSI CUSTOMER_26_DSI
    DSO CUSTOMER_DSO
     USING(551,572)
     ALLOCATE PRIME    ON SP51 SIZE 264004K
                          SP52 SIZE 264000K,
          OVERFLOW ON SP51 SIZE 13201K
                          SP52 SIZE 13200K;

CREATE DSI CUSTOMER_27_DSI
    DSO CUSTOMER_DSO
     USING(573,594)
     ALLOCATE PRIME    ON SP53 SIZE 264004K
                          SP54 SIZE 264000K,
          OVERFLOW ON SP53 SIZE 13201K
                          SP54 SIZE 13200K;

CREATE DSI CUSTOMER_28_DSI
    DSO CUSTOMER_DSO
     USING(595,616)
     ALLOCATE PRIME    ON SP55 SIZE 264004K
                          SP56 SIZE 264000K,
          OVERFLOW ON SP55 SIZE 13201K
                          SP56 SIZE 13200K;

CREATE DSI CUSTOMER_29_DSI
    DSO CUSTOMER_DSO
     USING(617,638)
     ALLOCATE PRIME    ON SP57 SIZE 264004K
                          SP58 SIZE 264000K,
          OVERFLOW ON SP57 SIZE 13201K
                          SP58 SIZE 13200K;

CREATE DSI CUSTOMER_30_DSI
    DSO CUSTOMER_DSO
     USING(639,660)
     ALLOCATE PRIME    ON SP59 SIZE 264004K
                          SP60 SIZE 264000K,
          OVERFLOW ON SP59 SIZE 13201K
                          SP60 SIZE 13200K;

CREATE DSI CUSTOMER_31_DSI
    DSO CUSTOMER_DSO
     USING(661,682)
     ALLOCATE PRIME    ON SP61 SIZE 264004K
                          SP62 SIZE 264000K,
          OVERFLOW ON SP61 SIZE 13201K
                          SP62 SIZE 13200K;

CREATE DSI CUSTOMER_32_DSI
    DSO CUSTOMER_DSO
     USING(683,704)
     ALLOCATE PRIME    ON SP63 SIZE 264004K
                          SP64 SIZE 264000K,
          OVERFLOW ON SP63 SIZE 13201K
                          SP64 SIZE 13200K;

CREATE DSI CUSTOMER_33_DSI
    DSO CUSTOMER_DSO
     USING(705,726)
     ALLOCATE PRIME    ON SP65 SIZE 264004K
                          SP66 SIZE 264000K,
          OVERFLOW ON SP65 SIZE 13201K
                          SP66 SIZE 13200K;

CREATE DSI CUSTOMER_34_DSI
    DSO CUSTOMER_DSO
     USING(727,748)
     ALLOCATE PRIME    ON SP67 SIZE 264004K
                          SP68 SIZE 264000K,
          OVERFLOW ON SP67 SIZE 13201K
```

SP68 SIZE 13200K;

CREATE DSI CUSTOMER_35_DSI
    DSO CUSTOMER_DSO
     USING(749,770)
     ALLOCATE PRIME   ON SP69 SIZE 264004K
                   SP70 SIZE 264000K,
         OVERFLOW ON SP69 SIZE 13201K
                   SP70 SIZE 13200K;

CREATE DSI CUSTOMER_36_DSI
    DSO CUSTOMER_DSO
     USING(771,792)
     ALLOCATE PRIME   ON SP71 SIZE 264004K
                   SP72 SIZE 264000K,
         OVERFLOW ON SP71 SIZE 13201K
                   SP72 SIZE 13200K;

CREATE DSI CUSTOMER_37_DSI
    DSO CUSTOMER_DSO
     USING(793,814)
     ALLOCATE PRIME   ON SP73 SIZE 264004K
                   SP74 SIZE 264000K,
         OVERFLOW ON SP73 SIZE 13201K
                   SP74 SIZE 13200K;

CREATE DSI CUSTOMER_38_DSI
    DSO CUSTOMER_DSO
     USING(815,836)
     ALLOCATE PRIME   ON SP75 SIZE 264004K
                   SP76 SIZE 264000K,
         OVERFLOW ON SP75 SIZE 13201K
                   SP76 SIZE 13200K;

CREATE DSI CUSTOMER_39_DSI
    DSO CUSTOMER_DSO
     USING(837,858)
     ALLOCATE PRIME   ON SP77 SIZE 264004K
                   SP78 SIZE 264000K,
         OVERFLOW ON SP77 SIZE 13201K
                   SP78 SIZE 13200K;

CREATE DSI CUSTOMER_40_DSI
    DSO CUSTOMER_DSO
     USING(859,880)
     ALLOCATE PRIME   ON SP79 SIZE 264004K
                   SP80 SIZE 264000K,
         OVERFLOW ON SP79 SIZE 13201K
                   SP80 SIZE 13200K;

CREATE DSI CUSTOMER_41_DSI
    DSO CUSTOMER_DSO
     USING(881,902)
     ALLOCATE PRIME   ON SP81 SIZE 264004K
                   SP82 SIZE 264000K,
         OVERFLOW ON SP81 SIZE 13201K
                   SP82 SIZE 13200K;

CREATE DSI CUSTOMER_42_DSI
    DSO CUSTOMER_DSO
     USING(903,924)
     ALLOCATE PRIME   ON SP83 SIZE 264004K
                   SP84 SIZE 264000K,

OVERFLOW ON SP83 SIZE 13201K
                   SP84 SIZE 13200K;

CREATE DSI CUSTOMER_43_DSI
    DSO CUSTOMER_DSO
     USING(925,946)
     ALLOCATE PRIME   ON SP85 SIZE 264004K
                   SP86 SIZE 264000K,
         OVERFLOW ON SP85 SIZE 13201K
                   SP86 SIZE 13200K;

CREATE DSI CUSTOMER_44_DSI
    DSO CUSTOMER_DSO
     USING(947,968)
     ALLOCATE PRIME   ON SP87 SIZE 264004K
                   SP88 SIZE 264000K,
         OVERFLOW ON SP87 SIZE 13201K
                   SP88 SIZE 13200K;

CREATE DSI CUSTOMER_45_DSI
    DSO CUSTOMER_DSO
     USING(969,990)
     ALLOCATE PRIME   ON SP89 SIZE 264004K
                   SP90 SIZE 264000K,
         OVERFLOW ON SP89 SIZE 13201K
                   SP90 SIZE 13200K;

CREATE DSI CUSTOMER_46_DSI
    DSO CUSTOMER_DSO
     USING(991,1012)
     ALLOCATE PRIME   ON SP91 SIZE 264004K
                   SP92 SIZE 264000K,
         OVERFLOW ON SP91 SIZE 13201K
                   SP92 SIZE 13200K;

CREATE DSI CUSTOMER_47_DSI
    DSO CUSTOMER_DSO
     USING(1013,1034)
     ALLOCATE PRIME   ON SP93 SIZE 264004K
                   SP94 SIZE 264000K,
         OVERFLOW ON SP93 SIZE 13201K
                   SP94 SIZE 13200K;

CREATE DSI CUSTOMER_48_DSI
    DSO CUSTOMER_DSO
     USING(1035,1056)
     ALLOCATE PRIME   ON SP95 SIZE 264004K
                   SP96 SIZE 264000K,
         OVERFLOW ON SP95 SIZE 13201K
                   SP96 SIZE 13200K;

CREATE DSI CUSTOMER_49_DSI
    DSO CUSTOMER_DSO
     USING(1057,1078)
     ALLOCATE PRIME   ON SP97 SIZE 264004K
                   SP98 SIZE 264000K,
         OVERFLOW ON SP97 SIZE 13201K
                   SP98 SIZE 13200K;

CREATE DSI CUSTOMER_50_DSI
    DSO CUSTOMER_DSO
     USING(1079,1100)
     ALLOCATE PRIME   ON SP99 SIZE 264004K

```
                              SP100 SIZE 264000K,
          OVERFLOW ON SP99 SIZE 13201K
                              SP100 SIZE 13200K;

CREATE DSI CUSTOMER_51_DSI
     DSO CUSTOMER_DSO
     USING(1101,1122)
     ALLOCATE PRIME    ON SP101 SIZE 264004K
                              SP102 SIZE 264000K,
          OVERFLOW ON SP101 SIZE 13201K
                              SP102 SIZE 13200K;

CREATE DSI CUSTOMER_52_DSI
     DSO CUSTOMER_DSO
     USING(1123,1144)
     ALLOCATE PRIME    ON SP103 SIZE 264004K
                              SP104 SIZE 264000K,
          OVERFLOW ON SP103 SIZE 13201K
                              SP104 SIZE 13200K;

CREATE DSI CUSTOMER_53_DSI
     DSO CUSTOMER_DSO
     USING(1145,1166)
     ALLOCATE PRIME    ON SP105 SIZE 264004K
                              SP106 SIZE 264000K,
          OVERFLOW ON SP105 SIZE 13201K
                              SP106 SIZE 13200K;

CREATE DSI CUSTOMER_54_DSI
     DSO CUSTOMER_DSO
     USING(1167,1188)
     ALLOCATE PRIME    ON SP107 SIZE 264004K
                              SP108 SIZE 264000K,
          OVERFLOW ON SP107 SIZE 13201K
                              SP108 SIZE 13200K;

CREATE DSI CUSTOMER_55_DSI
     DSO CUSTOMER_DSO
     USING(1189,1210)
     ALLOCATE PRIME    ON SP109 SIZE 264004K
                              SP110 SIZE 264000K,
          OVERFLOW ON SP109 SIZE 13201K
                              SP110 SIZE 13200K;

CREATE DSI CUSTOMER_56_DSI
     DSO CUSTOMER_DSO
     USING(1211,1232)
     ALLOCATE PRIME    ON SP111 SIZE 264004K
                              SP112 SIZE 264000K,
          OVERFLOW ON SP111 SIZE 13201K
                              SP112 SIZE 13200K;

CREATE DSI CUSTOMER_57_DSI
     DSO CUSTOMER_DSO
     USING(1233,1254)
     ALLOCATE PRIME    ON SP113 SIZE 264004K
                              SP114 SIZE 264000K,
          OVERFLOW ON SP113 SIZE 13201K
                              SP114 SIZE 13200K;

CREATE DSI CUSTOMER_58_DSI
     DSO CUSTOMER_DSO
     USING(1255,1276)
```

```
     ALLOCATE PRIME    ON SP115 SIZE 264004K
                              SP116 SIZE 264000K,
          OVERFLOW ON SP115 SIZE 13201K
                              SP116 SIZE 13200K;

CREATE DSI CUSTOMER_59_DSI
     DSO CUSTOMER_DSO
     USING(1277,1298)
     ALLOCATE PRIME    ON SP117 SIZE 264004K
                              SP118 SIZE 264000K,
          OVERFLOW ON SP117 SIZE 13201K
                              SP118 SIZE 13200K;

CREATE DSI CUSTOMER_60_DSI
     DSO CUSTOMER_DSO
     USING(1299,1320)
     ALLOCATE PRIME    ON SP119 SIZE 264004K
                              SP120 SIZE 264000K,
          OVERFLOW ON SP119 SIZE 13201K
                              SP120 SIZE 13200K;

CREATE DSI CUSTOMER_61_DSI
     DSO CUSTOMER_DSO
     USING(1321,1342)
     ALLOCATE PRIME    ON SP121 SIZE 264004K
                              SP122 SIZE 264000K,
          OVERFLOW ON SP121 SIZE 13201K
                              SP122 SIZE 13200K;

CREATE DSI CUSTOMER_62_DSI
     DSO CUSTOMER_DSO
     USING(1343,1364)
     ALLOCATE PRIME    ON SP123 SIZE 264004K
                              SP124 SIZE 264000K,
          OVERFLOW ON SP123 SIZE 13201K
                              SP124 SIZE 13200K;

CREATE DSI CUSTOMER_63_DSI
     DSO CUSTOMER_DSO
     USING(1365,1386)
     ALLOCATE PRIME    ON SP125 SIZE 264004K
                              SP126 SIZE 264000K,
          OVERFLOW ON SP125 SIZE 13201K
                              SP126 SIZE 13200K;

CREATE DSI CUSTOMER_64_DSI
     DSO CUSTOMER_DSO
     USING(1387,1408)
     ALLOCATE PRIME    ON SP127 SIZE 264004K
                              SP128 SIZE 264000K,
          OVERFLOW ON SP127 SIZE 13201K
                              SP128 SIZE 13200K;

CREATE DSI CUSTOMER_65_DSI
     DSO CUSTOMER_DSO
     USING(1409,1430)
     ALLOCATE PRIME    ON SP129 SIZE 264004K
                              SP130 SIZE 264000K,
          OVERFLOW ON SP129 SIZE 13201K
                              SP130 SIZE 13200K;

CREATE DSI CUSTOMER_66_DSI
     DSO CUSTOMER_DSO
```

```
        USING(1431,1452)
        ALLOCATE PRIME    ON SP131 SIZE 264004K
                          SP132 SIZE 264000K,
            OVERFLOW ON SP131 SIZE 13201K
                          SP132 SIZE 13200K;

CREATE DSI CUSTOMER_67_DSI
    DSO CUSTOMER_DSO
        USING(1453,1474)
        ALLOCATE PRIME    ON SP133 SIZE 264004K
                          SP134 SIZE 264000K,
            OVERFLOW ON SP133 SIZE 13201K
                          SP134 SIZE 13200K;

CREATE DSI CUSTOMER_68_DSI
    DSO CUSTOMER_DSO
        USING(1475,1496)
        ALLOCATE PRIME    ON SP135 SIZE 264004K
                          SP136 SIZE 264000K,
            OVERFLOW ON SP135 SIZE 13201K
                          SP136 SIZE 13200K;

CREATE DSI CUSTOMER_69_DSI
    DSO CUSTOMER_DSO
        USING(1497,1518)
        ALLOCATE PRIME    ON SP137 SIZE 264004K
                          SP138 SIZE 264000K,
            OVERFLOW ON SP137 SIZE 13201K
                          SP138 SIZE 13200K;

CREATE DSI CUSTOMER_70_DSI
    DSO CUSTOMER_DSO
        USING(1519,1540)
        ALLOCATE PRIME    ON SP139 SIZE 264004K
                          SP140 SIZE 264000K,
            OVERFLOW ON SP139 SIZE 13201K
                          SP140 SIZE 13200K;

CREATE DSI CUSTOMER_71_DSI
    DSO CUSTOMER_DSO
        USING(1541,1562)
        ALLOCATE PRIME    ON SP141 SIZE 264004K
                          SP142 SIZE 264000K,
            OVERFLOW ON SP141 SIZE 13201K
                          SP142 SIZE 13200K;

CREATE DSI CUSTOMER_72_DSI
    DSO CUSTOMER_DSO
        USING(1563,1584)
        ALLOCATE PRIME    ON SP143 SIZE 264004K
                          SP144 SIZE 264000K,
            OVERFLOW ON SP143 SIZE 13201K
                          SP144 SIZE 13200K;

CREATE DSI CUSTOMER_73_DSI
    DSO CUSTOMER_DSO
        USING(1585,1606)
        ALLOCATE PRIME    ON SP145 SIZE 264004K
                          SP146 SIZE 264000K,
            OVERFLOW ON SP145 SIZE 13201K
                          SP146 SIZE 13200K;

CREATE DSI CUSTOMER_74_DSI
```

```
    DSO CUSTOMER_DSO
        USING(1607,1628)
        ALLOCATE PRIME    ON SP147 SIZE 264004K
                          SP148 SIZE 264000K,
            OVERFLOW ON SP147 SIZE 13201K
                          SP148 SIZE 13200K;

CREATE DSI CUSTOMER_75_DSI
    DSO CUSTOMER_DSO
        USING(1629,1650)
        ALLOCATE PRIME    ON SP149 SIZE 264004K
                          SP150 SIZE 264000K,
            OVERFLOW ON SP149 SIZE 13201K
                          SP150 SIZE 13200K;

CREATE DSI CUSTOMER_76_DSI
    DSO CUSTOMER_DSO
        USING(1651,1672)
        ALLOCATE PRIME    ON SP151 SIZE 264004K
                          SP152 SIZE 264000K,
            OVERFLOW ON SP151 SIZE 13201K
                          SP152 SIZE 13200K;

CREATE DSI CUSTOMER_77_DSI
    DSO CUSTOMER_DSO
        USING(1673,1694)
        ALLOCATE PRIME    ON SP153 SIZE 264004K
                          SP154 SIZE 264000K,
            OVERFLOW ON SP153 SIZE 13201K
                          SP154 SIZE 13200K;

CREATE DSI CUSTOMER_78_DSI
    DSO CUSTOMER_DSO
        USING(1695,1716)
        ALLOCATE PRIME    ON SP155 SIZE 264004K
                          SP156 SIZE 264000K,
            OVERFLOW ON SP155 SIZE 13201K
                          SP156 SIZE 13200K;

CREATE DSI CUSTOMER_79_DSI
    DSO CUSTOMER_DSO
        USING(1717,1738)
        ALLOCATE PRIME    ON SP157 SIZE 264004K
                          SP158 SIZE 264000K,
            OVERFLOW ON SP157 SIZE 13201K
                          SP158 SIZE 13200K;

CREATE DSI CUSTOMER_80_DSI
    DSO CUSTOMER_DSO
        USING(1739,1760)
        ALLOCATE PRIME    ON SP159 SIZE 264004K
                          SP160 SIZE 264000K,
            OVERFLOW ON SP159 SIZE 13201K
                          SP160 SIZE 13200K;

CREATE DSI CUSTOMER_81_DSI
    DSO CUSTOMER_DSO
        USING(1761,1782)
        ALLOCATE PRIME    ON SP161 SIZE 264004K
                          SP162 SIZE 264000K,
            OVERFLOW ON SP161 SIZE 13201K
                          SP162 SIZE 13200K;
```

CREATE DSI CUSTOMER_82_DSI
    DSO CUSTOMER_DSO
    USING(1783,1804)
    ALLOCATE PRIME   ON SP163 SIZE 264004K
                   SP164 SIZE 264000K,
        OVERFLOW ON SP163 SIZE 13201K
                   SP164 SIZE 13200K;

CREATE DSI CUSTOMER_83_DSI
    DSO CUSTOMER_DSO
    USING(1805,1826)
    ALLOCATE PRIME   ON SP165 SIZE 264004K
                   SP166 SIZE 264000K,
        OVERFLOW ON SP165 SIZE 13201K
                   SP166 SIZE 13200K;

CREATE DSI CUSTOMER_84_DSI
    DSO CUSTOMER_DSO
    USING(1827,1848)
    ALLOCATE PRIME   ON SP167 SIZE 264004K
                   SP168 SIZE 264000K,
        OVERFLOW ON SP167 SIZE 13201K
                   SP168 SIZE 13200K;

CREATE DSI CUSTOMER_85_DSI
    DSO CUSTOMER_DSO
    USING(1849,1870)
    ALLOCATE PRIME   ON SP169 SIZE 264004K
                   SP170 SIZE 264000K,
        OVERFLOW ON SP169 SIZE 13201K
                   SP170 SIZE 13200K;

CREATE DSI CUSTOMER_86_DSI
    DSO CUSTOMER_DSO
    USING(1871,1892)
    ALLOCATE PRIME   ON SP171 SIZE 264004K
                   SP172 SIZE 264000K,
        OVERFLOW ON SP171 SIZE 13201K
                   SP172 SIZE 13200K;

CREATE DSI CUSTOMER_87_DSI
    DSO CUSTOMER_DSO
    USING(1893,1914)
    ALLOCATE PRIME   ON SP173 SIZE 264004K
                   SP174 SIZE 264000K,
        OVERFLOW ON SP173 SIZE 13201K
                   SP174 SIZE 13200K;

CREATE DSI CUSTOMER_88_DSI
    DSO CUSTOMER_DSO
    USING(1915,1936)
    ALLOCATE PRIME   ON SP175 SIZE 264004K
                   SP176 SIZE 264000K,
        OVERFLOW ON SP175 SIZE 13201K
                   SP176 SIZE 13200K;

CREATE DSI CUSTOMER_89_DSI
    DSO CUSTOMER_DSO
    USING(1937,1958)
    ALLOCATE PRIME   ON SP177 SIZE 264004K
                   SP178 SIZE 264000K,
        OVERFLOW ON SP177 SIZE 13201K
                   SP178 SIZE 13200K;

CREATE DSI CUSTOMER_90_DSI
    DSO CUSTOMER_DSO
    USING(1959,1980)
    ALLOCATE PRIME   ON SP179 SIZE 264004K
                   SP180 SIZE 264000K,
        OVERFLOW ON SP179 SIZE 13201K
                   SP180 SIZE 13200K;

CREATE DSI CUSTOMER_91_DSI
    DSO CUSTOMER_DSO
    USING(1981,2002)
    ALLOCATE PRIME   ON SP181 SIZE 264004K
                   SP182 SIZE 264000K,
        OVERFLOW ON SP181 SIZE 13201K
                   SP182 SIZE 13200K;

CREATE DSI CUSTOMER_92_DSI
    DSO CUSTOMER_DSO
    USING(2003,2024)
    ALLOCATE PRIME   ON SP183 SIZE 264004K
                   SP184 SIZE 264000K,
        OVERFLOW ON SP183 SIZE 13201K
                   SP184 SIZE 13200K;

CREATE DSI CUSTOMER_93_DSI
    DSO CUSTOMER_DSO
    USING(2025,2046)
    ALLOCATE PRIME   ON SP185 SIZE 264004K
                   SP186 SIZE 264000K,
        OVERFLOW ON SP185 SIZE 13201K
                   SP186 SIZE 13200K;

CREATE DSI CUSTOMER_94_DSI
    DSO CUSTOMER_DSO
    USING(2047,2068)
    ALLOCATE PRIME   ON SP187 SIZE 264004K
                   SP188 SIZE 264000K,
        OVERFLOW ON SP187 SIZE 13201K
                   SP188 SIZE 13200K;

CREATE DSI CUSTOMER_95_DSI
    DSO CUSTOMER_DSO
    USING(2069,2090)
    ALLOCATE PRIME   ON SP189 SIZE 264004K
                   SP190 SIZE 264000K,
        OVERFLOW ON SP189 SIZE 13201K
                   SP190 SIZE 13200K;

CREATE DSI CUSTOMER_96_DSI
    DSO CUSTOMER_DSO
    USING(2091,4224)
    ALLOCATE PRIME   ON SP191 SIZE 264004K
                   SP192 SIZE 264000K,
        OVERFLOW ON SP191 SIZE 13201K
                   SP192 SIZE 13200K;

:::::::::::::
ddl.dat.DI
:::::::::::::

-- * Phase.2-2: District

```
CREATE DSO DISTRICT_DSO
    FROM TPCC_SCHEMA.DISTRICT
        TYPE
RANDOM(PAGESIZE1(1),PAGESIZE2(1),RULE(D_W_ID*20+D_ID*2)
)
        WHERE (D_W_ID) BETWEEN (?) AND (?);

CREATE DSI DISTRICT_1_DSI
    DSO DISTRICT_DSO
     USING(1,88)
      ALLOCATE PRIME    ON SP1 SIZE 2349K,
           OVERFLOW ON SP1 SIZE 58K;

CREATE DSI DISTRICT_2_DSI
    DSO DISTRICT_DSO
     USING(89,176)
      ALLOCATE PRIME    ON SP7 SIZE 2349K,
           OVERFLOW ON SP7 SIZE 58K;

CREATE DSI DISTRICT_3_DSI
    DSO DISTRICT_DSO
     USING(177,264)
      ALLOCATE PRIME    ON SP13 SIZE 2349K,
           OVERFLOW ON SP13 SIZE 58K;

CREATE DSI DISTRICT_4_DSI
    DSO DISTRICT_DSO
     USING(265,352)
      ALLOCATE PRIME    ON SP19 SIZE 2349K,
           OVERFLOW ON SP19 SIZE 58K;

CREATE DSI DISTRICT_5_DSI
    DSO DISTRICT_DSO
     USING(353,440)
      ALLOCATE PRIME    ON SP25 SIZE 2349K,
           OVERFLOW ON SP25 SIZE 58K;

CREATE DSI DISTRICT_6_DSI
    DSO DISTRICT_DSO
     USING(441,528)
      ALLOCATE PRIME    ON SP31 SIZE 2349K,
           OVERFLOW ON SP31 SIZE 58K;

CREATE DSI DISTRICT_7_DSI
    DSO DISTRICT_DSO
     USING(529,616)
      ALLOCATE PRIME    ON SP37 SIZE 2349K,
           OVERFLOW ON SP37 SIZE 58K;

CREATE DSI DISTRICT_8_DSI
    DSO DISTRICT_DSO
     USING(617,704)
      ALLOCATE PRIME    ON SP43 SIZE 2349K,
           OVERFLOW ON SP43 SIZE 58K;

CREATE DSI DISTRICT_9_DSI
    DSO DISTRICT_DSO
     USING(705,792)
      ALLOCATE PRIME    ON SP49 SIZE 2349K,
           OVERFLOW ON SP49 SIZE 58K;

CREATE DSI DISTRICT_10_DSI
    DSO DISTRICT_DSO
     USING(793,880)
      ALLOCATE PRIME    ON SP55 SIZE 2349K,
           OVERFLOW ON SP55 SIZE 58K;

CREATE DSI DISTRICT_11_DSI
    DSO DISTRICT_DSO
     USING(881,968)
      ALLOCATE PRIME    ON SP61 SIZE 2349K,
           OVERFLOW ON SP61 SIZE 58K;

CREATE DSI DISTRICT_12_DSI
    DSO DISTRICT_DSO
     USING(969,1056)
      ALLOCATE PRIME    ON SP67 SIZE 2349K,
           OVERFLOW ON SP67 SIZE 58K;

CREATE DSI DISTRICT_13_DSI
    DSO DISTRICT_DSO
     USING(1057,1144)
      ALLOCATE PRIME    ON SP73 SIZE 2349K,
           OVERFLOW ON SP73 SIZE 58K;

CREATE DSI DISTRICT_14_DSI
    DSO DISTRICT_DSO
     USING(1145,1232)
      ALLOCATE PRIME    ON SP79 SIZE 2349K,
           OVERFLOW ON SP79 SIZE 58K;

CREATE DSI DISTRICT_15_DSI
    DSO DISTRICT_DSO
     USING(1233,1320)
      ALLOCATE PRIME    ON SP85 SIZE 2349K,
           OVERFLOW ON SP85 SIZE 58K;

CREATE DSI DISTRICT_16_DSI
    DSO DISTRICT_DSO
     USING(1321,1408)
      ALLOCATE PRIME    ON SP91 SIZE 2349K,
           OVERFLOW ON SP91 SIZE 58K;

CREATE DSI DISTRICT_17_DSI
    DSO DISTRICT_DSO
     USING(1409,1496)
      ALLOCATE PRIME    ON SP97 SIZE 2349K,
           OVERFLOW ON SP97 SIZE 58K;

CREATE DSI DISTRICT_18_DSI
    DSO DISTRICT_DSO
     USING(1497,1584)
      ALLOCATE PRIME    ON SP103 SIZE 2349K,
           OVERFLOW ON SP103 SIZE 58K;

CREATE DSI DISTRICT_19_DSI
    DSO DISTRICT_DSO
     USING(1585,1672)
      ALLOCATE PRIME    ON SP109 SIZE 2349K,
           OVERFLOW ON SP109 SIZE 58K;

CREATE DSI DISTRICT_20_DSI
    DSO DISTRICT_DSO
     USING(1673,1760)
      ALLOCATE PRIME    ON SP115 SIZE 2349K,
           OVERFLOW ON SP115 SIZE 58K;
```

CREATE DSI DISTRICT_21_DSI
    DSO DISTRICT_DSO
      USING(1761,1848)
      ALLOCATE PRIME   ON SP121 SIZE 2349K,
          OVERFLOW ON SP121 SIZE 58K;

CREATE DSI DISTRICT_22_DSI
    DSO DISTRICT_DSO
      USING(1849,1936)
      ALLOCATE PRIME   ON SP127 SIZE 2349K,
          OVERFLOW ON SP127 SIZE 58K;

CREATE DSI DISTRICT_23_DSI
    DSO DISTRICT_DSO
      USING(1937,2024)
      ALLOCATE PRIME   ON SP133 SIZE 2349K,
          OVERFLOW ON SP133 SIZE 58K;

CREATE DSI DISTRICT_24_DSI
    DSO DISTRICT_DSO
      USING(2025,4224)
      ALLOCATE PRIME   ON SP139 SIZE 2349K,
          OVERFLOW ON SP139 SIZE 58K;

::::::::::::::
ddl.dat.HI
::::::::::::::

-- * Phase.2-7: History

CREATE DSO HISTORY_DSO
    FROM TPCC_SCHEMA.HISTORY
        TYPE SEQUENTIAL(PAGESIZE(4),ORDER(0))
        WHERE (H_W_ID) BETWEEN (?) AND (?);

CREATE DSI HISTORY_1_DSI
    DSO HISTORY_DSO
      USING(1,22)
      ALLOCATE DATA   ON SP1 SIZE 32444K
                 SP2 SIZE 32440K;

CREATE DSI HISTORY_2_DSI
    DSO HISTORY_DSO
      USING(23,44)
      ALLOCATE DATA   ON SP3 SIZE 32444K
                 SP4 SIZE 32440K;

CREATE DSI HISTORY_3_DSI
    DSO HISTORY_DSO
      USING(45,66)
      ALLOCATE DATA   ON SP5 SIZE 32444K
                 SP6 SIZE 32440K;

CREATE DSI HISTORY_4_DSI
    DSO HISTORY_DSO
      USING(67,88)
      ALLOCATE DATA   ON SP7 SIZE 32444K
                 SP8 SIZE 32440K;

CREATE DSI HISTORY_5_DSI
    DSO HISTORY_DSO
      USING(89,110)

ALLOCATE DATA   ON SP9 SIZE 32444K
                 SP10 SIZE 32440K;

CREATE DSI HISTORY_6_DSI
    DSO HISTORY_DSO
      USING(111,132)
      ALLOCATE DATA   ON SP11 SIZE 32444K
                 SP12 SIZE 32440K;

CREATE DSI HISTORY_7_DSI
    DSO HISTORY_DSO
      USING(133,154)
      ALLOCATE DATA   ON SP13 SIZE 32444K
                 SP14 SIZE 32440K;

CREATE DSI HISTORY_8_DSI
    DSO HISTORY_DSO
      USING(155,176)
      ALLOCATE DATA   ON SP15 SIZE 32444K
                 SP16 SIZE 32440K;

CREATE DSI HISTORY_9_DSI
    DSO HISTORY_DSO
      USING(177,198)
      ALLOCATE DATA   ON SP17 SIZE 32444K
                 SP18 SIZE 32440K;

CREATE DSI HISTORY_10_DSI
    DSO HISTORY_DSO
      USING(199,220)
      ALLOCATE DATA   ON SP19 SIZE 32444K
                 SP20 SIZE 32440K;

CREATE DSI HISTORY_11_DSI
    DSO HISTORY_DSO
      USING(221,242)
      ALLOCATE DATA   ON SP21 SIZE 32444K
                 SP22 SIZE 32440K;

CREATE DSI HISTORY_12_DSI
    DSO HISTORY_DSO
      USING(243,264)
      ALLOCATE DATA   ON SP23 SIZE 32444K
                 SP24 SIZE 32440K;

CREATE DSI HISTORY_13_DSI
    DSO HISTORY_DSO
      USING(265,286)
      ALLOCATE DATA   ON SP25 SIZE 32444K
                 SP26 SIZE 32440K;

CREATE DSI HISTORY_14_DSI
    DSO HISTORY_DSO
      USING(287,308)
      ALLOCATE DATA   ON SP27 SIZE 32444K
                 SP28 SIZE 32440K;

CREATE DSI HISTORY_15_DSI
    DSO HISTORY_DSO
      USING(309,330)
      ALLOCATE DATA   ON SP29 SIZE 32444K
                 SP30 SIZE 32440K;

CREATE DSI HISTORY_16_DSI
    DSO HISTORY_DSO
    USING(331,352)
    ALLOCATE DATA   ON SP31 SIZE 32444K
                     SP32 SIZE 32440K;

CREATE DSI HISTORY_17_DSI
    DSO HISTORY_DSO
    USING(353,374)
    ALLOCATE DATA   ON SP33 SIZE 32444K
                     SP34 SIZE 32440K;

CREATE DSI HISTORY_18_DSI
    DSO HISTORY_DSO
    USING(375,396)
    ALLOCATE DATA   ON SP35 SIZE 32444K
                     SP36 SIZE 32440K;

CREATE DSI HISTORY_19_DSI
    DSO HISTORY_DSO
    USING(397,418)
    ALLOCATE DATA   ON SP37 SIZE 32444K
                     SP38 SIZE 32440K;

CREATE DSI HISTORY_20_DSI
    DSO HISTORY_DSO
    USING(419,440)
    ALLOCATE DATA   ON SP39 SIZE 32444K
                     SP40 SIZE 32440K;

CREATE DSI HISTORY_21_DSI
    DSO HISTORY_DSO
    USING(441,462)
    ALLOCATE DATA   ON SP41 SIZE 32444K
                     SP42 SIZE 32440K;

CREATE DSI HISTORY_22_DSI
    DSO HISTORY_DSO
    USING(463,484)
    ALLOCATE DATA   ON SP43 SIZE 32444K
                     SP44 SIZE 32440K;

CREATE DSI HISTORY_23_DSI
    DSO HISTORY_DSO
    USING(485,506)
    ALLOCATE DATA   ON SP45 SIZE 32444K
                     SP46 SIZE 32440K;

CREATE DSI HISTORY_24_DSI
    DSO HISTORY_DSO
    USING(507,528)
    ALLOCATE DATA   ON SP47 SIZE 32444K
                     SP48 SIZE 32440K;

CREATE DSI HISTORY_25_DSI
    DSO HISTORY_DSO
    USING(529,550)
    ALLOCATE DATA   ON SP49 SIZE 32444K
                     SP50 SIZE 32440K;

CREATE DSI HISTORY_26_DSI
    DSO HISTORY_DSO
    USING(551,572)

    ALLOCATE DATA   ON SP51 SIZE 32444K
                     SP52 SIZE 32440K;

CREATE DSI HISTORY_27_DSI
    DSO HISTORY_DSO
    USING(573,594)
    ALLOCATE DATA   ON SP53 SIZE 32444K
                     SP54 SIZE 32440K;

CREATE DSI HISTORY_28_DSI
    DSO HISTORY_DSO
    USING(595,616)
    ALLOCATE DATA   ON SP55 SIZE 32444K
                     SP56 SIZE 32440K;

CREATE DSI HISTORY_29_DSI
    DSO HISTORY_DSO
    USING(617,638)
    ALLOCATE DATA   ON SP57 SIZE 32444K
                     SP58 SIZE 32440K;

CREATE DSI HISTORY_30_DSI
    DSO HISTORY_DSO
    USING(639,660)
    ALLOCATE DATA   ON SP59 SIZE 32444K
                     SP60 SIZE 32440K;

CREATE DSI HISTORY_31_DSI
    DSO HISTORY_DSO
    USING(661,682)
    ALLOCATE DATA   ON SP61 SIZE 32444K
                     SP62 SIZE 32440K;

CREATE DSI HISTORY_32_DSI
    DSO HISTORY_DSO
    USING(683,704)
    ALLOCATE DATA   ON SP63 SIZE 32444K
                     SP64 SIZE 32440K;

CREATE DSI HISTORY_33_DSI
    DSO HISTORY_DSO
    USING(705,726)
    ALLOCATE DATA   ON SP65 SIZE 32444K
                     SP66 SIZE 32440K;

CREATE DSI HISTORY_34_DSI
    DSO HISTORY_DSO
    USING(727,748)
    ALLOCATE DATA   ON SP67 SIZE 32444K
                     SP68 SIZE 32440K;

CREATE DSI HISTORY_35_DSI
    DSO HISTORY_DSO
    USING(749,770)
    ALLOCATE DATA   ON SP69 SIZE 32444K
                     SP70 SIZE 32440K;

CREATE DSI HISTORY_36_DSI
    DSO HISTORY_DSO
    USING(771,792)
    ALLOCATE DATA   ON SP71 SIZE 32444K
                     SP72 SIZE 32440K;

CREATE DSI HISTORY_37_DSI
    DSO HISTORY_DSO
    USING(793,814)
    ALLOCATE DATA   ON SP73 SIZE 32444K
                    SP74 SIZE 32440K;

CREATE DSI HISTORY_38_DSI
    DSO HISTORY_DSO
    USING(815,836)
    ALLOCATE DATA   ON SP75 SIZE 32444K
                    SP76 SIZE 32440K;

CREATE DSI HISTORY_39_DSI
    DSO HISTORY_DSO
    USING(837,858)
    ALLOCATE DATA   ON SP77 SIZE 32444K
                    SP78 SIZE 32440K;

CREATE DSI HISTORY_40_DSI
    DSO HISTORY_DSO
    USING(859,880)
    ALLOCATE DATA   ON SP79 SIZE 32444K
                    SP80 SIZE 32440K;

CREATE DSI HISTORY_41_DSI
    DSO HISTORY_DSO
    USING(881,902)
    ALLOCATE DATA   ON SP81 SIZE 32444K
                    SP82 SIZE 32440K;

CREATE DSI HISTORY_42_DSI
    DSO HISTORY_DSO
    USING(903,924)
    ALLOCATE DATA   ON SP83 SIZE 32444K
                    SP84 SIZE 32440K;

CREATE DSI HISTORY_43_DSI
    DSO HISTORY_DSO
    USING(925,946)
    ALLOCATE DATA   ON SP85 SIZE 32444K
                    SP86 SIZE 32440K;

CREATE DSI HISTORY_44_DSI
    DSO HISTORY_DSO
    USING(947,968)
    ALLOCATE DATA   ON SP87 SIZE 32444K
                    SP88 SIZE 32440K;

CREATE DSI HISTORY_45_DSI
    DSO HISTORY_DSO
    USING(969,990)
    ALLOCATE DATA   ON SP89 SIZE 32444K
                    SP90 SIZE 32440K;

CREATE DSI HISTORY_46_DSI
    DSO HISTORY_DSO
    USING(991,1012)
    ALLOCATE DATA   ON SP91 SIZE 32444K
                    SP92 SIZE 32440K;

CREATE DSI HISTORY_47_DSI
    DSO HISTORY_DSO
    USING(1013,1034)

    ALLOCATE DATA   ON SP93 SIZE 32444K
                    SP94 SIZE 32440K;

CREATE DSI HISTORY_48_DSI
    DSO HISTORY_DSO
    USING(1035,1056)
    ALLOCATE DATA   ON SP95 SIZE 32444K
                    SP96 SIZE 32440K;

CREATE DSI HISTORY_49_DSI
    DSO HISTORY_DSO
    USING(1057,1078)
    ALLOCATE DATA   ON SP97 SIZE 32444K
                    SP98 SIZE 32440K;

CREATE DSI HISTORY_50_DSI
    DSO HISTORY_DSO
    USING(1079,1100)
    ALLOCATE DATA   ON SP99 SIZE 32444K
                    SP100 SIZE 32440K;

CREATE DSI HISTORY_51_DSI
    DSO HISTORY_DSO
    USING(1101,1122)
    ALLOCATE DATA   ON SP101 SIZE 32444K
                    SP102 SIZE 32440K;

CREATE DSI HISTORY_52_DSI
    DSO HISTORY_DSO
    USING(1123,1144)
    ALLOCATE DATA   ON SP103 SIZE 32444K
                    SP104 SIZE 32440K;

CREATE DSI HISTORY_53_DSI
    DSO HISTORY_DSO
    USING(1145,1166)
    ALLOCATE DATA   ON SP105 SIZE 32444K
                    SP106 SIZE 32440K;

CREATE DSI HISTORY_54_DSI
    DSO HISTORY_DSO
    USING(1167,1188)
    ALLOCATE DATA   ON SP107 SIZE 32444K
                    SP108 SIZE 32440K;

CREATE DSI HISTORY_55_DSI
    DSO HISTORY_DSO
    USING(1189,1210)
    ALLOCATE DATA   ON SP109 SIZE 32444K
                    SP110 SIZE 32440K;

CREATE DSI HISTORY_56_DSI
    DSO HISTORY_DSO
    USING(1211,1232)
    ALLOCATE DATA   ON SP111 SIZE 32444K
                    SP112 SIZE 32440K;

CREATE DSI HISTORY_57_DSI
    DSO HISTORY_DSO
    USING(1233,1254)
    ALLOCATE DATA   ON SP113 SIZE 32444K
                    SP114 SIZE 32440K;

```
CREATE DSI HISTORY_58_DSI
    DSO HISTORY_DSO
    USING(1255,1276)
    ALLOCATE DATA      ON SP115 SIZE 32444K
                          SP116 SIZE 32440K;

CREATE DSI HISTORY_59_DSI
    DSO HISTORY_DSO
    USING(1277,1298)
    ALLOCATE DATA      ON SP117 SIZE 32444K
                          SP118 SIZE 32440K;

CREATE DSI HISTORY_60_DSI
    DSO HISTORY_DSO
    USING(1299,1320)
    ALLOCATE DATA      ON SP119 SIZE 32444K
                          SP120 SIZE 32440K;

CREATE DSI HISTORY_61_DSI
    DSO HISTORY_DSO
    USING(1321,1342)
    ALLOCATE DATA      ON SP121 SIZE 32444K
                          SP122 SIZE 32440K;

CREATE DSI HISTORY_62_DSI
    DSO HISTORY_DSO
    USING(1343,1364)
    ALLOCATE DATA      ON SP123 SIZE 32444K
                          SP124 SIZE 32440K;

CREATE DSI HISTORY_63_DSI
    DSO HISTORY_DSO
    USING(1365,1386)
    ALLOCATE DATA      ON SP125 SIZE 32444K
                          SP126 SIZE 32440K;

CREATE DSI HISTORY_64_DSI
    DSO HISTORY_DSO
    USING(1387,1408)
    ALLOCATE DATA      ON SP127 SIZE 32444K
                          SP128 SIZE 32440K;

CREATE DSI HISTORY_65_DSI
    DSO HISTORY_DSO
    USING(1409,1430)
    ALLOCATE DATA      ON SP129 SIZE 32444K
                          SP130 SIZE 32440K;

CREATE DSI HISTORY_66_DSI
    DSO HISTORY_DSO
    USING(1431,1452)
    ALLOCATE DATA      ON SP131 SIZE 32444K
                          SP132 SIZE 32440K;

CREATE DSI HISTORY_67_DSI
    DSO HISTORY_DSO
    USING(1453,1474)
    ALLOCATE DATA      ON SP133 SIZE 32444K
                          SP134 SIZE 32440K;

CREATE DSI HISTORY_68_DSI
    DSO HISTORY_DSO
    USING(1475,1496)
```

```
    ALLOCATE DATA      ON SP135 SIZE 32444K
                          SP136 SIZE 32440K;

CREATE DSI HISTORY_69_DSI
    DSO HISTORY_DSO
    USING(1497,1518)
    ALLOCATE DATA      ON SP137 SIZE 32444K
                          SP138 SIZE 32440K;

CREATE DSI HISTORY_70_DSI
    DSO HISTORY_DSO
    USING(1519,1540)
    ALLOCATE DATA      ON SP139 SIZE 32444K
                          SP140 SIZE 32440K;

CREATE DSI HISTORY_71_DSI
    DSO HISTORY_DSO
    USING(1541,1562)
    ALLOCATE DATA      ON SP141 SIZE 32444K
                          SP142 SIZE 32440K;

CREATE DSI HISTORY_72_DSI
    DSO HISTORY_DSO
    USING(1563,1584)
    ALLOCATE DATA      ON SP143 SIZE 32444K
                          SP144 SIZE 32440K;

CREATE DSI HISTORY_73_DSI
    DSO HISTORY_DSO
    USING(1585,1606)
    ALLOCATE DATA      ON SP145 SIZE 32444K
                          SP146 SIZE 32440K;

CREATE DSI HISTORY_74_DSI
    DSO HISTORY_DSO
    USING(1607,1628)
    ALLOCATE DATA      ON SP147 SIZE 32444K
                          SP148 SIZE 32440K;

CREATE DSI HISTORY_75_DSI
    DSO HISTORY_DSO
    USING(1629,1650)
    ALLOCATE DATA      ON SP149 SIZE 32444K
                          SP150 SIZE 32440K;

CREATE DSI HISTORY_76_DSI
    DSO HISTORY_DSO
    USING(1651,1672)
    ALLOCATE DATA      ON SP151 SIZE 32444K
                          SP152 SIZE 32440K;

CREATE DSI HISTORY_77_DSI
    DSO HISTORY_DSO
    USING(1673,1694)
    ALLOCATE DATA      ON SP153 SIZE 32444K
                          SP154 SIZE 32440K;

CREATE DSI HISTORY_78_DSI
    DSO HISTORY_DSO
    USING(1695,1716)
    ALLOCATE DATA      ON SP155 SIZE 32444K
                          SP156 SIZE 32440K;
```

```
CREATE DSI HISTORY_79_DSI
    DSO HISTORY_DSO
    USING(1717,1738)
    ALLOCATE DATA      ON SP157 SIZE 32444K
                          SP158 SIZE 32440K;

CREATE DSI HISTORY_80_DSI
    DSO HISTORY_DSO
    USING(1739,1760)
    ALLOCATE DATA      ON SP159 SIZE 32444K
                          SP160 SIZE 32440K;

CREATE DSI HISTORY_81_DSI
    DSO HISTORY_DSO
    USING(1761,1782)
    ALLOCATE DATA      ON SP161 SIZE 32444K
                          SP162 SIZE 32440K;

CREATE DSI HISTORY_82_DSI
    DSO HISTORY_DSO
    USING(1783,1804)
    ALLOCATE DATA      ON SP163 SIZE 32444K
                          SP164 SIZE 32440K;

CREATE DSI HISTORY_83_DSI
    DSO HISTORY_DSO
    USING(1805,1826)
    ALLOCATE DATA      ON SP165 SIZE 32444K
                          SP166 SIZE 32440K;

CREATE DSI HISTORY_84_DSI
    DSO HISTORY_DSO
    USING(1827,1848)
    ALLOCATE DATA      ON SP167 SIZE 32444K
                          SP168 SIZE 32440K;

CREATE DSI HISTORY_85_DSI
    DSO HISTORY_DSO
    USING(1849,1870)
    ALLOCATE DATA      ON SP169 SIZE 32444K
                          SP170 SIZE 32440K;

CREATE DSI HISTORY_86_DSI
    DSO HISTORY_DSO
    USING(1871,1892)
    ALLOCATE DATA      ON SP171 SIZE 32444K
                          SP172 SIZE 32440K;

CREATE DSI HISTORY_87_DSI
    DSO HISTORY_DSO
    USING(1893,1914)
    ALLOCATE DATA      ON SP173 SIZE 32444K
                          SP174 SIZE 32440K;

CREATE DSI HISTORY_88_DSI
    DSO HISTORY_DSO
    USING(1915,1936)
    ALLOCATE DATA      ON SP175 SIZE 32444K
                          SP176 SIZE 32440K;

CREATE DSI HISTORY_89_DSI
    DSO HISTORY_DSO
    USING(1937,1958)

    ALLOCATE DATA      ON SP177 SIZE 32444K
                          SP178 SIZE 32440K;

CREATE DSI HISTORY_90_DSI
    DSO HISTORY_DSO
    USING(1959,1980)
    ALLOCATE DATA      ON SP179 SIZE 32444K
                          SP180 SIZE 32440K;

CREATE DSI HISTORY_91_DSI
    DSO HISTORY_DSO
    USING(1981,2002)
    ALLOCATE DATA      ON SP181 SIZE 32444K
                          SP182 SIZE 32440K;

CREATE DSI HISTORY_92_DSI
    DSO HISTORY_DSO
    USING(2003,2024)
    ALLOCATE DATA      ON SP183 SIZE 32444K
                          SP184 SIZE 32440K;

CREATE DSI HISTORY_93_DSI
    DSO HISTORY_DSO
    USING(2025,2046)
    ALLOCATE DATA      ON SP185 SIZE 32444K
                          SP186 SIZE 32440K;

CREATE DSI HISTORY_94_DSI
    DSO HISTORY_DSO
    USING(2047,2068)
    ALLOCATE DATA      ON SP187 SIZE 32444K
                          SP188 SIZE 32440K;

CREATE DSI HISTORY_95_DSI
    DSO HISTORY_DSO
    USING(2069,2090)
    ALLOCATE DATA      ON SP189 SIZE 32444K
                          SP190 SIZE 32440K;

CREATE DSI HISTORY_96_DSI
    DSO HISTORY_DSO
    USING(2091,4224)
    ALLOCATE DATA      ON SP191 SIZE 32444K
                          SP192 SIZE 32440K;

:::::::::::::
ddl.dat.IT
:::::::::::::
---------------------------------------------------------------------------
-- * Phase.2-9: Item
---------------------------------------------------------------------------
    CREATE DSO ITEM_DSO
        FROM TPCC_SCHEMA.ITEM
        TYPE
RANDOM(PAGESIZE1(1),PAGESIZE2(1),RULE(I_ID/7+(I_ID-
((I_ID/7)*7))*14286));

    CREATE DSI ITEM_1_DSI
        DSO ITEM_DSO
        ALLOCATE PRIME    ONSP1 SIZE 1435K
                             SP2 SIZE 1428K
                             SP3 SIZE 1428K
                             SP4 SIZE 1428K
```

                                    SP5 SIZE 1428K
                                    SP6 SIZE 1428K
                                    SP7 SIZE 1428K
                                    SP8 SIZE 1428K
                                    SP9 SIZE 1428K
                                    SP10 SIZE 1428K,
                    OVERFLOW ONSP1 SIZE 716K;

::::::::::::::
ddl.dat.NI
::::::::::::::

-- * Phase.2-6b: NewOrder-Index

        CREATE DSO NEWORDER_IX_DSO
            INDEX ON
TPCC_SCHEMA.NEWORDER(NO_W_ID,NO_D_ID,NO_O_ID)
            TYPE
BTREE(PAGESIZE1(8),PAGESIZE2(32),DEGENERATE);

        CREATE DSI NEWORDER_X_1_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_1_DSI
            ALLOCATE INDEX   ON SP2 SIZE 256K,
                BASE   ON SP1 SIZE 6624K
                                    SP2 SIZE 6616K;

        CREATE DSI NEWORDER_X_2_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_2_DSI
            ALLOCATE INDEX   ON SP4 SIZE 256K,
                BASE   ON SP3 SIZE 6624K
                                    SP4 SIZE 6616K;

        CREATE DSI NEWORDER_X_3_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_3_DSI
            ALLOCATE INDEX   ON SP6 SIZE 256K,
                BASE   ON SP5 SIZE 6624K
                                    SP6 SIZE 6616K;

        CREATE DSI NEWORDER_X_4_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_4_DSI
            ALLOCATE INDEX   ON SP8 SIZE 256K,
                BASE   ON SP7 SIZE 6624K
                                    SP8 SIZE 6616K;

        CREATE DSI NEWORDER_X_5_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_5_DSI
            ALLOCATE INDEX   ON SP10 SIZE 256K,
                BASE   ON SP9 SIZE 6624K
                                    SP10 SIZE 6616K;

        CREATE DSI NEWORDER_X_6_DSI
            INDEX
            DSO NEWORDER_IX_DSO

            BASE NEWORDER_6_DSI
            ALLOCATE INDEX   ON SP12 SIZE 256K,
                BASE   ON SP11 SIZE 6624K
                                    SP12 SIZE 6616K;

CREATE DSI NEWORDER_X_7_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_7_DSI
            ALLOCATE INDEX   ON SP14 SIZE 256K,
                BASE   ON SP13 SIZE 6624K
                                    SP14 SIZE 6616K;

CREATE DSI NEWORDER_X_8_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_8_DSI
            ALLOCATE INDEX   ON SP16 SIZE 256K,
                BASE   ON SP15 SIZE 6624K
                                    SP16 SIZE 6616K;

CREATE DSI NEWORDER_X_9_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_9_DSI
            ALLOCATE INDEX   ON SP18 SIZE 256K,
                BASE   ON SP17 SIZE 6624K
                                    SP18 SIZE 6616K;

CREATE DSI NEWORDER_X_10_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_10_DSI
            ALLOCATE INDEX   ON SP20 SIZE 256K,
                BASE   ON SP19 SIZE 6624K
                                    SP20 SIZE 6616K;

CREATE DSI NEWORDER_X_11_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_11_DSI
            ALLOCATE INDEX   ON SP22 SIZE 256K,
                BASE   ON SP21 SIZE 6624K
                                    SP22 SIZE 6616K;

CREATE DSI NEWORDER_X_12_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_12_DSI
            ALLOCATE INDEX   ON SP24 SIZE 256K,
                BASE   ON SP23 SIZE 6624K
                                    SP24 SIZE 6616K;

CREATE DSI NEWORDER_X_13_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_13_DSI
            ALLOCATE INDEX   ON SP26 SIZE 256K,
                BASE   ON SP25 SIZE 6624K
                                    SP26 SIZE 6616K;

CREATE DSI NEWORDER_X_14_DSI
            INDEX

```
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_14_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP28 SIZE 256K,                  BASE NEWORDER_22_DSI
             BASE    ON SP27 SIZE 6624K                       ALLOCATE INDEX    ON SP44 SIZE 256K,
                          SP28 SIZE 6616K;                         BASE    ON SP43 SIZE 6624K
                                                                                 SP44 SIZE 6616K;
CREATE DSI NEWORDER_X_15_DSI
        INDEX                                          CREATE DSI NEWORDER_X_23_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_15_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP30 SIZE 256K,                  BASE NEWORDER_23_DSI
             BASE    ON SP29 SIZE 6624K                       ALLOCATE INDEX    ON SP46 SIZE 256K,
                          SP30 SIZE 6616K;                         BASE    ON SP45 SIZE 6624K
                                                                                 SP46 SIZE 6616K;
CREATE DSI NEWORDER_X_16_DSI
        INDEX                                          CREATE DSI NEWORDER_X_24_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_16_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP32 SIZE 256K,                  BASE NEWORDER_24_DSI
             BASE    ON SP31 SIZE 6624K                       ALLOCATE INDEX    ON SP48 SIZE 256K,
                          SP32 SIZE 6616K;                         BASE    ON SP47 SIZE 6624K
                                                                                 SP48 SIZE 6616K;
CREATE DSI NEWORDER_X_17_DSI
        INDEX                                          CREATE DSI NEWORDER_X_25_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_17_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP34 SIZE 256K,                  BASE NEWORDER_25_DSI
             BASE    ON SP33 SIZE 6624K                       ALLOCATE INDEX    ON SP50 SIZE 256K,
                          SP34 SIZE 6616K;                         BASE    ON SP49 SIZE 6624K
                                                                                 SP50 SIZE 6616K;
CREATE DSI NEWORDER_X_18_DSI
        INDEX                                          CREATE DSI NEWORDER_X_26_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_18_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP36 SIZE 256K,                  BASE NEWORDER_26_DSI
             BASE    ON SP35 SIZE 6624K                       ALLOCATE INDEX    ON SP52 SIZE 256K,
                          SP36 SIZE 6616K;                         BASE    ON SP51 SIZE 6624K
                                                                                 SP52 SIZE 6616K;
CREATE DSI NEWORDER_X_19_DSI
        INDEX                                          CREATE DSI NEWORDER_X_27_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_19_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP38 SIZE 256K,                  BASE NEWORDER_27_DSI
             BASE    ON SP37 SIZE 6624K                       ALLOCATE INDEX    ON SP54 SIZE 256K,
                          SP38 SIZE 6616K;                         BASE    ON SP53 SIZE 6624K
                                                                                 SP54 SIZE 6616K;
CREATE DSI NEWORDER_X_20_DSI
        INDEX                                          CREATE DSI NEWORDER_X_28_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_20_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP40 SIZE 256K,                  BASE NEWORDER_28_DSI
             BASE    ON SP39 SIZE 6624K                       ALLOCATE INDEX    ON SP56 SIZE 256K,
                          SP40 SIZE 6616K;                         BASE    ON SP55 SIZE 6624K
                                                                                 SP56 SIZE 6616K;
CREATE DSI NEWORDER_X_21_DSI
        INDEX                                          CREATE DSI NEWORDER_X_29_DSI
        DSO NEWORDER_IX_DSO                                   INDEX
        BASE NEWORDER_21_DSI                                  DSO NEWORDER_IX_DSO
        ALLOCATE INDEX    ON SP42 SIZE 256K,                  BASE NEWORDER_29_DSI
             BASE    ON SP41 SIZE 6624K                       ALLOCATE INDEX    ON SP58 SIZE 256K,
                          SP42 SIZE 6616K;                         BASE    ON SP57 SIZE 6624K
                                                                                 SP58 SIZE 6616K;
CREATE DSI NEWORDER_X_22_DSI
```

CREATE DSI NEWORDER_X_30_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_30_DSI
    ALLOCATE INDEX   ON SP60 SIZE 256K,
        BASE   ON SP59 SIZE 6624K
                    SP60 SIZE 6616K;

CREATE DSI NEWORDER_X_31_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_31_DSI
    ALLOCATE INDEX   ON SP62 SIZE 256K,
        BASE   ON SP61 SIZE 6624K
                    SP62 SIZE 6616K;

CREATE DSI NEWORDER_X_32_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_32_DSI
    ALLOCATE INDEX   ON SP64 SIZE 256K,
        BASE   ON SP63 SIZE 6624K
                    SP64 SIZE 6616K;

CREATE DSI NEWORDER_X_33_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_33_DSI
    ALLOCATE INDEX   ON SP66 SIZE 256K,
        BASE   ON SP65 SIZE 6624K
                    SP66 SIZE 6616K;

CREATE DSI NEWORDER_X_34_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_34_DSI
    ALLOCATE INDEX   ON SP68 SIZE 256K,
        BASE   ON SP67 SIZE 6624K
                    SP68 SIZE 6616K;

CREATE DSI NEWORDER_X_35_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_35_DSI
    ALLOCATE INDEX   ON SP70 SIZE 256K,
        BASE   ON SP69 SIZE 6624K
                    SP70 SIZE 6616K;

CREATE DSI NEWORDER_X_36_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_36_DSI
    ALLOCATE INDEX   ON SP72 SIZE 256K,
        BASE   ON SP71 SIZE 6624K
                    SP72 SIZE 6616K;

CREATE DSI NEWORDER_X_37_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_37_DSI
    ALLOCATE INDEX   ON SP74 SIZE 256K,
        BASE   ON SP73 SIZE 6624K
                    SP74 SIZE 6616K;

CREATE DSI NEWORDER_X_38_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_38_DSI
    ALLOCATE INDEX   ON SP76 SIZE 256K,
        BASE   ON SP75 SIZE 6624K
                    SP76 SIZE 6616K;

CREATE DSI NEWORDER_X_39_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_39_DSI
    ALLOCATE INDEX   ON SP78 SIZE 256K,
        BASE   ON SP77 SIZE 6624K
                    SP78 SIZE 6616K;

CREATE DSI NEWORDER_X_40_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_40_DSI
    ALLOCATE INDEX   ON SP80 SIZE 256K,
        BASE   ON SP79 SIZE 6624K
                    SP80 SIZE 6616K;

CREATE DSI NEWORDER_X_41_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_41_DSI
    ALLOCATE INDEX   ON SP82 SIZE 256K,
        BASE   ON SP81 SIZE 6624K
                    SP82 SIZE 6616K;

CREATE DSI NEWORDER_X_42_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_42_DSI
    ALLOCATE INDEX   ON SP84 SIZE 256K,
        BASE   ON SP83 SIZE 6624K
                    SP84 SIZE 6616K;

CREATE DSI NEWORDER_X_43_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_43_DSI
    ALLOCATE INDEX   ON SP86 SIZE 256K,
        BASE   ON SP85 SIZE 6624K
                    SP86 SIZE 6616K;

CREATE DSI NEWORDER_X_44_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_44_DSI
    ALLOCATE INDEX   ON SP88 SIZE 256K,
        BASE   ON SP87 SIZE 6624K
                    SP88 SIZE 6616K;

CREATE DSI NEWORDER_X_45_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_45_DSI
    ALLOCATE INDEX   ON SP90 SIZE 256K,
        BASE   ON SP89 SIZE 6624K

```
                            SP90 SIZE 6616K;                                    BASE     ON SP105 SIZE 6624K
                                                                                            SP106 SIZE 6616K;
CREATE DSI NEWORDER_X_46_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_54_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_46_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP92 SIZE 256K,                         BASE NEWORDER_54_DSI
         BASE     ON SP91 SIZE 6624K                              ALLOCATE INDEX    ON SP108 SIZE 256K,
                            SP92 SIZE 6616K;                          BASE     ON SP107 SIZE 6624K
                                                                                            SP108 SIZE 6616K;
CREATE DSI NEWORDER_X_47_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_55_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_47_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP94 SIZE 256K,                         BASE NEWORDER_55_DSI
         BASE     ON SP93 SIZE 6624K                              ALLOCATE INDEX    ON SP110 SIZE 256K,
                            SP94 SIZE 6616K;                          BASE     ON SP109 SIZE 6624K
                                                                                            SP110 SIZE 6616K;
CREATE DSI NEWORDER_X_48_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_56_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_48_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP96 SIZE 256K,                         BASE NEWORDER_56_DSI
         BASE     ON SP95 SIZE 6624K                              ALLOCATE INDEX    ON SP112 SIZE 256K,
                            SP96 SIZE 6616K;                          BASE     ON SP111 SIZE 6624K
                                                                                            SP112 SIZE 6616K;
CREATE DSI NEWORDER_X_49_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_57_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_49_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP98 SIZE 256K,                         BASE NEWORDER_57_DSI
         BASE     ON SP97 SIZE 6624K                              ALLOCATE INDEX    ON SP114 SIZE 256K,
                            SP98 SIZE 6616K;                          BASE     ON SP113 SIZE 6624K
                                                                                            SP114 SIZE 6616K;
CREATE DSI NEWORDER_X_50_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_58_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_50_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP100 SIZE 256K,                        BASE NEWORDER_58_DSI
         BASE     ON SP99 SIZE 6624K                              ALLOCATE INDEX    ON SP116 SIZE 256K,
                            SP100 SIZE 6616K;                         BASE     ON SP115 SIZE 6624K
                                                                                            SP116 SIZE 6616K;
CREATE DSI NEWORDER_X_51_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_59_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_51_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP102 SIZE 256K,                        BASE NEWORDER_59_DSI
         BASE     ON SP101 SIZE 6624K                              ALLOCATE INDEX    ON SP118 SIZE 256K,
                            SP102 SIZE 6616K;                         BASE     ON SP117 SIZE 6624K
                                                                                            SP118 SIZE 6616K;
CREATE DSI NEWORDER_X_52_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_60_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_52_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP104 SIZE 256K,                        BASE NEWORDER_60_DSI
         BASE     ON SP103 SIZE 6624K                              ALLOCATE INDEX    ON SP120 SIZE 256K,
                            SP104 SIZE 6616K;                         BASE     ON SP119 SIZE 6624K
                                                                                            SP120 SIZE 6616K;
CREATE DSI NEWORDER_X_53_DSI
     INDEX                                                   CREATE DSI NEWORDER_X_61_DSI
     DSO NEWORDER_IX_DSO                                          INDEX
     BASE NEWORDER_53_DSI                                         DSO NEWORDER_IX_DSO
     ALLOCATE INDEX    ON SP106 SIZE 256K,                        BASE NEWORDER_61_DSI
```

ALLOCATE INDEX   ON SP122 SIZE 256K,
        BASE   ON SP121 SIZE 6624K
                        SP122 SIZE 6616K;

CREATE DSI NEWORDER_X_62_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_62_DSI
    ALLOCATE INDEX   ON SP124 SIZE 256K,
        BASE   ON SP123 SIZE 6624K
                        SP124 SIZE 6616K;

CREATE DSI NEWORDER_X_63_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_63_DSI
    ALLOCATE INDEX   ON SP126 SIZE 256K,
        BASE   ON SP125 SIZE 6624K
                        SP126 SIZE 6616K;

CREATE DSI NEWORDER_X_64_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_64_DSI
    ALLOCATE INDEX   ON SP128 SIZE 256K,
        BASE   ON SP127 SIZE 6624K
                        SP128 SIZE 6616K;

CREATE DSI NEWORDER_X_65_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_65_DSI
    ALLOCATE INDEX   ON SP130 SIZE 256K,
        BASE   ON SP129 SIZE 6624K
                        SP130 SIZE 6616K;

CREATE DSI NEWORDER_X_66_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_66_DSI
    ALLOCATE INDEX   ON SP132 SIZE 256K,
        BASE   ON SP131 SIZE 6624K
                        SP132 SIZE 6616K;

CREATE DSI NEWORDER_X_67_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_67_DSI
    ALLOCATE INDEX   ON SP134 SIZE 256K,
        BASE   ON SP133 SIZE 6624K
                        SP134 SIZE 6616K;

CREATE DSI NEWORDER_X_68_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_68_DSI
    ALLOCATE INDEX   ON SP136 SIZE 256K,
        BASE   ON SP135 SIZE 6624K
                        SP136 SIZE 6616K;

CREATE DSI NEWORDER_X_69_DSI
    INDEX
    DSO NEWORDER_IX_DSO

BASE NEWORDER_69_DSI
    ALLOCATE INDEX   ON SP138 SIZE 256K,
        BASE   ON SP137 SIZE 6624K
                        SP138 SIZE 6616K;

CREATE DSI NEWORDER_X_70_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_70_DSI
    ALLOCATE INDEX   ON SP140 SIZE 256K,
        BASE   ON SP139 SIZE 6624K
                        SP140 SIZE 6616K;

CREATE DSI NEWORDER_X_71_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_71_DSI
    ALLOCATE INDEX   ON SP142 SIZE 256K,
        BASE   ON SP141 SIZE 6624K
                        SP142 SIZE 6616K;

CREATE DSI NEWORDER_X_72_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_72_DSI
    ALLOCATE INDEX   ON SP144 SIZE 256K,
        BASE   ON SP143 SIZE 6624K
                        SP144 SIZE 6616K;

CREATE DSI NEWORDER_X_73_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_73_DSI
    ALLOCATE INDEX   ON SP146 SIZE 256K,
        BASE   ON SP145 SIZE 6624K
                        SP146 SIZE 6616K;

CREATE DSI NEWORDER_X_74_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_74_DSI
    ALLOCATE INDEX   ON SP148 SIZE 256K,
        BASE   ON SP147 SIZE 6624K
                        SP148 SIZE 6616K;

CREATE DSI NEWORDER_X_75_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_75_DSI
    ALLOCATE INDEX   ON SP150 SIZE 256K,
        BASE   ON SP149 SIZE 6624K
                        SP150 SIZE 6616K;

CREATE DSI NEWORDER_X_76_DSI
    INDEX
    DSO NEWORDER_IX_DSO
    BASE NEWORDER_76_DSI
    ALLOCATE INDEX   ON SP152 SIZE 256K,
        BASE   ON SP151 SIZE 6624K
                        SP152 SIZE 6616K;

CREATE DSI NEWORDER_X_77_DSI
    INDEX

```
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_77_DSI
        ALLOCATE INDEX    ON SP154 SIZE 256K,
            BASE     ON SP153 SIZE 6624K
                               SP154 SIZE 6616K;

CREATE DSI NEWORDER_X_78_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_78_DSI
        ALLOCATE INDEX    ON SP156 SIZE 256K,
            BASE     ON SP155 SIZE 6624K
                               SP156 SIZE 6616K;

CREATE DSI NEWORDER_X_79_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_79_DSI
        ALLOCATE INDEX    ON SP158 SIZE 256K,
            BASE     ON SP157 SIZE 6624K
                               SP158 SIZE 6616K;

CREATE DSI NEWORDER_X_80_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_80_DSI
        ALLOCATE INDEX    ON SP160 SIZE 256K,
            BASE     ON SP159 SIZE 6624K
                               SP160 SIZE 6616K;

CREATE DSI NEWORDER_X_81_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_81_DSI
        ALLOCATE INDEX    ON SP162 SIZE 256K,
            BASE     ON SP161 SIZE 6624K
                               SP162 SIZE 6616K;

CREATE DSI NEWORDER_X_82_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_82_DSI
        ALLOCATE INDEX    ON SP164 SIZE 256K,
            BASE     ON SP163 SIZE 6624K
                               SP164 SIZE 6616K;

CREATE DSI NEWORDER_X_83_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_83_DSI
        ALLOCATE INDEX    ON SP166 SIZE 256K,
            BASE     ON SP165 SIZE 6624K
                               SP166 SIZE 6616K;

CREATE DSI NEWORDER_X_84_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_84_DSI
        ALLOCATE INDEX    ON SP168 SIZE 256K,
            BASE     ON SP167 SIZE 6624K
                               SP168 SIZE 6616K;

CREATE DSI NEWORDER_X_85_DSI
```

```
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_85_DSI
        ALLOCATE INDEX    ON SP170 SIZE 256K,
            BASE     ON SP169 SIZE 6624K
                               SP170 SIZE 6616K;

CREATE DSI NEWORDER_X_86_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_86_DSI
        ALLOCATE INDEX    ON SP172 SIZE 256K,
            BASE     ON SP171 SIZE 6624K
                               SP172 SIZE 6616K;

CREATE DSI NEWORDER_X_87_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_87_DSI
        ALLOCATE INDEX    ON SP174 SIZE 256K,
            BASE     ON SP173 SIZE 6624K
                               SP174 SIZE 6616K;

CREATE DSI NEWORDER_X_88_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_88_DSI
        ALLOCATE INDEX    ON SP176 SIZE 256K,
            BASE     ON SP175 SIZE 6624K
                               SP176 SIZE 6616K;

CREATE DSI NEWORDER_X_89_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_89_DSI
        ALLOCATE INDEX    ON SP178 SIZE 256K,
            BASE     ON SP177 SIZE 6624K
                               SP178 SIZE 6616K;

CREATE DSI NEWORDER_X_90_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_90_DSI
        ALLOCATE INDEX    ON SP180 SIZE 256K,
            BASE     ON SP179 SIZE 6624K
                               SP180 SIZE 6616K;

CREATE DSI NEWORDER_X_91_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_91_DSI
        ALLOCATE INDEX    ON SP182 SIZE 256K,
            BASE     ON SP181 SIZE 6624K
                               SP182 SIZE 6616K;

CREATE DSI NEWORDER_X_92_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_92_DSI
        ALLOCATE INDEX    ON SP184 SIZE 256K,
            BASE     ON SP183 SIZE 6624K
                               SP184 SIZE 6616K;
```

```
    CREATE DSI NEWORDER_X_93_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_93_DSI
        ALLOCATE INDEX    ON SP186 SIZE 256K,
            BASE     ON SP185 SIZE 6624K
                              SP186 SIZE 6616K;

    CREATE DSI NEWORDER_X_94_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_94_DSI
        ALLOCATE INDEX    ON SP188 SIZE 256K,
            BASE     ON SP187 SIZE 6624K
                              SP188 SIZE 6616K;

    CREATE DSI NEWORDER_X_95_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_95_DSI
        ALLOCATE INDEX    ON SP190 SIZE 256K,
            BASE     ON SP189 SIZE 6624K
                              SP190 SIZE 6616K;

    CREATE DSI NEWORDER_X_96_DSI
        INDEX
        DSO NEWORDER_IX_DSO
        BASE NEWORDER_96_DSI
        ALLOCATE INDEX    ON SP192 SIZE 256K,
            BASE     ON SP191 SIZE 6624K
                              SP192 SIZE 6616K;

:::::::::::::
ddl.dat.NO
:::::::::::::
--------------------------------------------------------------------------
-- * Phase.2-6a: NewOrder
--------------------------------------------------------------------------
    CREATE DSO NEWORDER_DSO
        FROM TPCC_SCHEMA.NEWORDER
        TYPE
RANDOM(PAGESIZE1(8),PAGESIZE2(1),RULE((NO_O_ID∕8)*22+N
O_W_ID+((NO_D_ID-1)*10+(NO_O_ID-((NO_O_ID∕8)*8)))*3586))
            WHERE (NO_W_ID) BETWEEN (?) AND (?);

    CREATE DSI NEWORDER_1_DSI
        DSO NEWORDER_DSO
        USING(1,22)
        ALLOCATE PRIME    ON SP1 SIZE 14352K
                              SP2 SIZE 14344K,
            OVERFLOW ON SP1 SIZE 501K
                              SP2 SIZE 500K;

    CREATE DSI NEWORDER_2_DSI
        DSO NEWORDER_DSO
        USING(23,44)
        ALLOCATE PRIME    ON SP3 SIZE 14352K
                              SP4 SIZE 14344K,
            OVERFLOW ON SP3 SIZE 501K
                              SP4 SIZE 500K;

    CREATE DSI NEWORDER_3_DSI
        DSO NEWORDER_DSO
```

```
        USING(45,66)
        ALLOCATE PRIME    ON SP5 SIZE 14352K
                              SP6 SIZE 14344K,
            OVERFLOW ON SP5 SIZE 501K
                              SP6 SIZE 500K;

CREATE DSI NEWORDER_4_DSI
    DSO NEWORDER_DSO
    USING(67,88)
    ALLOCATE PRIME    ON SP7 SIZE 14352K
                          SP8 SIZE 14344K,
        OVERFLOW ON SP7 SIZE 501K
                          SP8 SIZE 500K;

CREATE DSI NEWORDER_5_DSI
    DSO NEWORDER_DSO
    USING(89,110)
    ALLOCATE PRIME    ON SP9 SIZE 14352K
                          SP10 SIZE 14344K,
        OVERFLOW ON SP9 SIZE 501K
                          SP10 SIZE 500K;

CREATE DSI NEWORDER_6_DSI
    DSO NEWORDER_DSO
    USING(111,132)
    ALLOCATE PRIME    ON SP11 SIZE 14352K
                          SP12 SIZE 14344K,
        OVERFLOW ON SP11 SIZE 501K
                          SP12 SIZE 500K;

CREATE DSI NEWORDER_7_DSI
    DSO NEWORDER_DSO
    USING(133,154)
    ALLOCATE PRIME    ON SP13 SIZE 14352K
                          SP14 SIZE 14344K,
        OVERFLOW ON SP13 SIZE 501K
                          SP14 SIZE 500K;

CREATE DSI NEWORDER_8_DSI
    DSO NEWORDER_DSO
    USING(155,176)
    ALLOCATE PRIME    ON SP15 SIZE 14352K
                          SP16 SIZE 14344K,
        OVERFLOW ON SP15 SIZE 501K
                          SP16 SIZE 500K;

CREATE DSI NEWORDER_9_DSI
    DSO NEWORDER_DSO
    USING(177,198)
    ALLOCATE PRIME    ON SP17 SIZE 14352K
                          SP18 SIZE 14344K,
        OVERFLOW ON SP17 SIZE 501K
                          SP18 SIZE 500K;

CREATE DSI NEWORDER_10_DSI
    DSO NEWORDER_DSO
    USING(199,220)
    ALLOCATE PRIME    ON SP19 SIZE 14352K
                          SP20 SIZE 14344K,
        OVERFLOW ON SP19 SIZE 501K
                          SP20 SIZE 500K;

CREATE DSI NEWORDER_11_DSI
```

```
        DSO NEWORDER_DSO
         USING(221,242)
          ALLOCATE PRIME    ON SP21 SIZE 14352K
                            SP22 SIZE 14344K,
              OVERFLOW ON SP21 SIZE 501K
                            SP22 SIZE 500K;

CREATE DSI NEWORDER_12_DSI
        DSO NEWORDER_DSO
         USING(243,264)
          ALLOCATE PRIME    ON SP23 SIZE 14352K
                            SP24 SIZE 14344K,
              OVERFLOW ON SP23 SIZE 501K
                            SP24 SIZE 500K;

CREATE DSI NEWORDER_13_DSI
        DSO NEWORDER_DSO
         USING(265,286)
          ALLOCATE PRIME    ON SP25 SIZE 14352K
                            SP26 SIZE 14344K,
              OVERFLOW ON SP25 SIZE 501K
                            SP26 SIZE 500K;

CREATE DSI NEWORDER_14_DSI
        DSO NEWORDER_DSO
         USING(287,308)
          ALLOCATE PRIME    ON SP27 SIZE 14352K
                            SP28 SIZE 14344K,
              OVERFLOW ON SP27 SIZE 501K
                            SP28 SIZE 500K;

CREATE DSI NEWORDER_15_DSI
        DSO NEWORDER_DSO
         USING(309,330)
          ALLOCATE PRIME    ON SP29 SIZE 14352K
                            SP30 SIZE 14344K,
              OVERFLOW ON SP29 SIZE 501K
                            SP30 SIZE 500K;

CREATE DSI NEWORDER_16_DSI
        DSO NEWORDER_DSO
         USING(331,352)
          ALLOCATE PRIME    ON SP31 SIZE 14352K
                            SP32 SIZE 14344K,
              OVERFLOW ON SP31 SIZE 501K
                            SP32 SIZE 500K;

CREATE DSI NEWORDER_17_DSI
        DSO NEWORDER_DSO
         USING(353,374)
          ALLOCATE PRIME    ON SP33 SIZE 14352K
                            SP34 SIZE 14344K,
              OVERFLOW ON SP33 SIZE 501K
                            SP34 SIZE 500K;

CREATE DSI NEWORDER_18_DSI
        DSO NEWORDER_DSO
         USING(375,396)
          ALLOCATE PRIME    ON SP35 SIZE 14352K
                            SP36 SIZE 14344K,
              OVERFLOW ON SP35 SIZE 501K
                            SP36 SIZE 500K;


CREATE DSI NEWORDER_19_DSI
        DSO NEWORDER_DSO
         USING(397,418)
          ALLOCATE PRIME    ON SP37 SIZE 14352K
                            SP38 SIZE 14344K,
              OVERFLOW ON SP37 SIZE 501K
                            SP38 SIZE 500K;

CREATE DSI NEWORDER_20_DSI
        DSO NEWORDER_DSO
         USING(419,440)
          ALLOCATE PRIME    ON SP39 SIZE 14352K
                            SP40 SIZE 14344K,
              OVERFLOW ON SP39 SIZE 501K
                            SP40 SIZE 500K;

CREATE DSI NEWORDER_21_DSI
        DSO NEWORDER_DSO
         USING(441,462)
          ALLOCATE PRIME    ON SP41 SIZE 14352K
                            SP42 SIZE 14344K,
              OVERFLOW ON SP41 SIZE 501K
                            SP42 SIZE 500K;

CREATE DSI NEWORDER_22_DSI
        DSO NEWORDER_DSO
         USING(463,484)
          ALLOCATE PRIME    ON SP43 SIZE 14352K
                            SP44 SIZE 14344K,
              OVERFLOW ON SP43 SIZE 501K
                            SP44 SIZE 500K;

CREATE DSI NEWORDER_23_DSI
        DSO NEWORDER_DSO
         USING(485,506)
          ALLOCATE PRIME    ON SP45 SIZE 14352K
                            SP46 SIZE 14344K,
              OVERFLOW ON SP45 SIZE 501K
                            SP46 SIZE 500K;

CREATE DSI NEWORDER_24_DSI
        DSO NEWORDER_DSO
         USING(507,528)
          ALLOCATE PRIME    ON SP47 SIZE 14352K
                            SP48 SIZE 14344K,
              OVERFLOW ON SP47 SIZE 501K
                            SP48 SIZE 500K;

CREATE DSI NEWORDER_25_DSI
        DSO NEWORDER_DSO
         USING(529,550)
          ALLOCATE PRIME    ON SP49 SIZE 14352K
                            SP50 SIZE 14344K,
              OVERFLOW ON SP49 SIZE 501K
                            SP50 SIZE 500K;

CREATE DSI NEWORDER_26_DSI
        DSO NEWORDER_DSO
         USING(551,572)
          ALLOCATE PRIME    ON SP51 SIZE 14352K
                            SP52 SIZE 14344K,
              OVERFLOW ON SP51 SIZE 501K
                            SP52 SIZE 500K;
```

SP68 SIZE 500K;

CREATE DSI NEWORDER_27_DSI
    DSO NEWORDER_DSO
     USING(573,594)
      ALLOCATE PRIME   ON SP53 SIZE 14352K
                         SP54 SIZE 14344K,
        OVERFLOW ON SP53 SIZE 501K
                         SP54 SIZE 500K;

CREATE DSI NEWORDER_28_DSI
    DSO NEWORDER_DSO
     USING(595,616)
      ALLOCATE PRIME   ON SP55 SIZE 14352K
                         SP56 SIZE 14344K,
        OVERFLOW ON SP55 SIZE 501K
                         SP56 SIZE 500K;

CREATE DSI NEWORDER_29_DSI
    DSO NEWORDER_DSO
     USING(617,638)
      ALLOCATE PRIME   ON SP57 SIZE 14352K
                         SP58 SIZE 14344K,
        OVERFLOW ON SP57 SIZE 501K
                         SP58 SIZE 500K;

CREATE DSI NEWORDER_30_DSI
    DSO NEWORDER_DSO
     USING(639,660)
      ALLOCATE PRIME   ON SP59 SIZE 14352K
                         SP60 SIZE 14344K,
        OVERFLOW ON SP59 SIZE 501K
                         SP60 SIZE 500K;

CREATE DSI NEWORDER_31_DSI
    DSO NEWORDER_DSO
     USING(661,682)
      ALLOCATE PRIME   ON SP61 SIZE 14352K
                         SP62 SIZE 14344K,
        OVERFLOW ON SP61 SIZE 501K
                         SP62 SIZE 500K;

CREATE DSI NEWORDER_32_DSI
    DSO NEWORDER_DSO
     USING(683,704)
      ALLOCATE PRIME   ON SP63 SIZE 14352K
                         SP64 SIZE 14344K,
        OVERFLOW ON SP63 SIZE 501K
                         SP64 SIZE 500K;

CREATE DSI NEWORDER_33_DSI
    DSO NEWORDER_DSO
     USING(705,726)
      ALLOCATE PRIME   ON SP65 SIZE 14352K
                         SP66 SIZE 14344K,
        OVERFLOW ON SP65 SIZE 501K
                         SP66 SIZE 500K;

CREATE DSI NEWORDER_34_DSI
    DSO NEWORDER_DSO
     USING(727,748)
      ALLOCATE PRIME   ON SP67 SIZE 14352K
                         SP68 SIZE 14344K,
        OVERFLOW ON SP67 SIZE 501K

CREATE DSI NEWORDER_35_DSI
    DSO NEWORDER_DSO
     USING(749,770)
      ALLOCATE PRIME   ON SP69 SIZE 14352K
                         SP70 SIZE 14344K,
        OVERFLOW ON SP69 SIZE 501K
                         SP70 SIZE 500K;

CREATE DSI NEWORDER_36_DSI
    DSO NEWORDER_DSO
     USING(771,792)
      ALLOCATE PRIME   ON SP71 SIZE 14352K
                         SP72 SIZE 14344K,
        OVERFLOW ON SP71 SIZE 501K
                         SP72 SIZE 500K;

CREATE DSI NEWORDER_37_DSI
    DSO NEWORDER_DSO
     USING(793,814)
      ALLOCATE PRIME   ON SP73 SIZE 14352K
                         SP74 SIZE 14344K,
        OVERFLOW ON SP73 SIZE 501K
                         SP74 SIZE 500K;

CREATE DSI NEWORDER_38_DSI
    DSO NEWORDER_DSO
     USING(815,836)
      ALLOCATE PRIME   ON SP75 SIZE 14352K
                         SP76 SIZE 14344K,
        OVERFLOW ON SP75 SIZE 501K
                         SP76 SIZE 500K;

CREATE DSI NEWORDER_39_DSI
    DSO NEWORDER_DSO
     USING(837,858)
      ALLOCATE PRIME   ON SP77 SIZE 14352K
                         SP78 SIZE 14344K,
        OVERFLOW ON SP77 SIZE 501K
                         SP78 SIZE 500K;

CREATE DSI NEWORDER_40_DSI
    DSO NEWORDER_DSO
     USING(859,880)
      ALLOCATE PRIME   ON SP79 SIZE 14352K
                         SP80 SIZE 14344K,
        OVERFLOW ON SP79 SIZE 501K
                         SP80 SIZE 500K;

CREATE DSI NEWORDER_41_DSI
    DSO NEWORDER_DSO
     USING(881,902)
      ALLOCATE PRIME   ON SP81 SIZE 14352K
                         SP82 SIZE 14344K,
        OVERFLOW ON SP81 SIZE 501K
                         SP82 SIZE 500K;

CREATE DSI NEWORDER_42_DSI
    DSO NEWORDER_DSO
     USING(903,924)
      ALLOCATE PRIME   ON SP83 SIZE 14352K
                         SP84 SIZE 14344K,

```
                         OVERFLOW ON SP83 SIZE 501K
                                      SP84 SIZE 500K;

CREATE DSI NEWORDER_43_DSI
     DSO NEWORDER_DSO
     USING(925,946)
     ALLOCATE PRIME    ON SP85 SIZE 14352K
                          SP86 SIZE 14344K,
            OVERFLOW ON SP85 SIZE 501K
                          SP86 SIZE 500K;

CREATE DSI NEWORDER_44_DSI
     DSO NEWORDER_DSO
     USING(947,968)
     ALLOCATE PRIME    ON SP87 SIZE 14352K
                          SP88 SIZE 14344K,
            OVERFLOW ON SP87 SIZE 501K
                          SP88 SIZE 500K;

CREATE DSI NEWORDER_45_DSI
     DSO NEWORDER_DSO
     USING(969,990)
     ALLOCATE PRIME    ON SP89 SIZE 14352K
                          SP90 SIZE 14344K,
            OVERFLOW ON SP89 SIZE 501K
                          SP90 SIZE 500K;

CREATE DSI NEWORDER_46_DSI
     DSO NEWORDER_DSO
     USING(991,1012)
     ALLOCATE PRIME    ON SP91 SIZE 14352K
                          SP92 SIZE 14344K,
            OVERFLOW ON SP91 SIZE 501K
                          SP92 SIZE 500K;

CREATE DSI NEWORDER_47_DSI
     DSO NEWORDER_DSO
     USING(1013,1034)
     ALLOCATE PRIME    ON SP93 SIZE 14352K
                          SP94 SIZE 14344K,
            OVERFLOW ON SP93 SIZE 501K
                          SP94 SIZE 500K;

CREATE DSI NEWORDER_48_DSI
     DSO NEWORDER_DSO
     USING(1035,1056)
     ALLOCATE PRIME    ON SP95 SIZE 14352K
                          SP96 SIZE 14344K,
            OVERFLOW ON SP95 SIZE 501K
                          SP96 SIZE 500K;

CREATE DSI NEWORDER_49_DSI
     DSO NEWORDER_DSO
     USING(1057,1078)
     ALLOCATE PRIME    ON SP97 SIZE 14352K
                          SP98 SIZE 14344K,
            OVERFLOW ON SP97 SIZE 501K
                          SP98 SIZE 500K;

CREATE DSI NEWORDER_50_DSI
     DSO NEWORDER_DSO
     USING(1079,1100)
     ALLOCATE PRIME    ON SP99 SIZE 14352K
```

```
                                     SP100 SIZE 14344K,
            OVERFLOW ON SP99 SIZE 501K
                          SP100 SIZE 500K;

CREATE DSI NEWORDER_51_DSI
     DSO NEWORDER_DSO
     USING(1101,1122)
     ALLOCATE PRIME    ON SP101 SIZE 14352K
                          SP102 SIZE 14344K,
            OVERFLOW ON SP101 SIZE 501K
                          SP102 SIZE 500K;

CREATE DSI NEWORDER_52_DSI
     DSO NEWORDER_DSO
     USING(1123,1144)
     ALLOCATE PRIME    ON SP103 SIZE 14352K
                          SP104 SIZE 14344K,
            OVERFLOW ON SP103 SIZE 501K
                          SP104 SIZE 500K;

CREATE DSI NEWORDER_53_DSI
     DSO NEWORDER_DSO
     USING(1145,1166)
     ALLOCATE PRIME    ON SP105 SIZE 14352K
                          SP106 SIZE 14344K,
            OVERFLOW ON SP105 SIZE 501K
                          SP106 SIZE 500K;

CREATE DSI NEWORDER_54_DSI
     DSO NEWORDER_DSO
     USING(1167,1188)
     ALLOCATE PRIME    ON SP107 SIZE 14352K
                          SP108 SIZE 14344K,
            OVERFLOW ON SP107 SIZE 501K
                          SP108 SIZE 500K;

CREATE DSI NEWORDER_55_DSI
     DSO NEWORDER_DSO
     USING(1189,1210)
     ALLOCATE PRIME    ON SP109 SIZE 14352K
                          SP110 SIZE 14344K,
            OVERFLOW ON SP109 SIZE 501K
                          SP110 SIZE 500K;

CREATE DSI NEWORDER_56_DSI
     DSO NEWORDER_DSO
     USING(1211,1232)
     ALLOCATE PRIME    ON SP111 SIZE 14352K
                          SP112 SIZE 14344K,
            OVERFLOW ON SP111 SIZE 501K
                          SP112 SIZE 500K;

CREATE DSI NEWORDER_57_DSI
     DSO NEWORDER_DSO
     USING(1233,1254)
     ALLOCATE PRIME    ON SP113 SIZE 14352K
                          SP114 SIZE 14344K,
            OVERFLOW ON SP113 SIZE 501K
                          SP114 SIZE 500K;

CREATE DSI NEWORDER_58_DSI
     DSO NEWORDER_DSO
     USING(1255,1276)
```

ALLOCATE PRIME   ON SP115 SIZE 14352K
                              SP116 SIZE 14344K,
              OVERFLOW ON SP115 SIZE 501K
                              SP116 SIZE 500K;

CREATE DSI NEWORDER_59_DSI
    DSO NEWORDER_DSO
    USING(1277,1298)
    ALLOCATE PRIME   ON SP117 SIZE 14352K
                              SP118 SIZE 14344K,
              OVERFLOW ON SP117 SIZE 501K
                              SP118 SIZE 500K;

CREATE DSI NEWORDER_60_DSI
    DSO NEWORDER_DSO
    USING(1299,1320)
    ALLOCATE PRIME   ON SP119 SIZE 14352K
                              SP120 SIZE 14344K,
              OVERFLOW ON SP119 SIZE 501K
                              SP120 SIZE 500K;

CREATE DSI NEWORDER_61_DSI
    DSO NEWORDER_DSO
    USING(1321,1342)
    ALLOCATE PRIME   ON SP121 SIZE 14352K
                              SP122 SIZE 14344K,
              OVERFLOW ON SP121 SIZE 501K
                              SP122 SIZE 500K;

CREATE DSI NEWORDER_62_DSI
    DSO NEWORDER_DSO
    USING(1343,1364)
    ALLOCATE PRIME   ON SP123 SIZE 14352K
                              SP124 SIZE 14344K,
              OVERFLOW ON SP123 SIZE 501K
                              SP124 SIZE 500K;

CREATE DSI NEWORDER_63_DSI
    DSO NEWORDER_DSO
    USING(1365,1386)
    ALLOCATE PRIME   ON SP125 SIZE 14352K
                              SP126 SIZE 14344K,
              OVERFLOW ON SP125 SIZE 501K
                              SP126 SIZE 500K;

CREATE DSI NEWORDER_64_DSI
    DSO NEWORDER_DSO
    USING(1387,1408)
    ALLOCATE PRIME   ON SP127 SIZE 14352K
                              SP128 SIZE 14344K,
              OVERFLOW ON SP127 SIZE 501K
                              SP128 SIZE 500K;

CREATE DSI NEWORDER_65_DSI
    DSO NEWORDER_DSO
    USING(1409,1430)
    ALLOCATE PRIME   ON SP129 SIZE 14352K
                              SP130 SIZE 14344K,
              OVERFLOW ON SP129 SIZE 501K
                              SP130 SIZE 500K;

CREATE DSI NEWORDER_66_DSI
    DSO NEWORDER_DSO

USING(1431,1452)
    ALLOCATE PRIME   ON SP131 SIZE 14352K
                              SP132 SIZE 14344K,
              OVERFLOW ON SP131 SIZE 501K
                              SP132 SIZE 500K;

CREATE DSI NEWORDER_67_DSI
    DSO NEWORDER_DSO
    USING(1453,1474)
    ALLOCATE PRIME   ON SP133 SIZE 14352K
                              SP134 SIZE 14344K,
              OVERFLOW ON SP133 SIZE 501K
                              SP134 SIZE 500K;

CREATE DSI NEWORDER_68_DSI
    DSO NEWORDER_DSO
    USING(1475,1496)
    ALLOCATE PRIME   ON SP135 SIZE 14352K
                              SP136 SIZE 14344K,
              OVERFLOW ON SP135 SIZE 501K
                              SP136 SIZE 500K;

CREATE DSI NEWORDER_69_DSI
    DSO NEWORDER_DSO
    USING(1497,1518)
    ALLOCATE PRIME   ON SP137 SIZE 14352K
                              SP138 SIZE 14344K,
              OVERFLOW ON SP137 SIZE 501K
                              SP138 SIZE 500K;

CREATE DSI NEWORDER_70_DSI
    DSO NEWORDER_DSO
    USING(1519,1540)
    ALLOCATE PRIME   ON SP139 SIZE 14352K
                              SP140 SIZE 14344K,
              OVERFLOW ON SP139 SIZE 501K
                              SP140 SIZE 500K;

CREATE DSI NEWORDER_71_DSI
    DSO NEWORDER_DSO
    USING(1541,1562)
    ALLOCATE PRIME   ON SP141 SIZE 14352K
                              SP142 SIZE 14344K,
              OVERFLOW ON SP141 SIZE 501K
                              SP142 SIZE 500K;

CREATE DSI NEWORDER_72_DSI
    DSO NEWORDER_DSO
    USING(1563,1584)
    ALLOCATE PRIME   ON SP143 SIZE 14352K
                              SP144 SIZE 14344K,
              OVERFLOW ON SP143 SIZE 501K
                              SP144 SIZE 500K;

CREATE DSI NEWORDER_73_DSI
    DSO NEWORDER_DSO
    USING(1585,1606)
    ALLOCATE PRIME   ON SP145 SIZE 14352K
                              SP146 SIZE 14344K,
              OVERFLOW ON SP145 SIZE 501K
                              SP146 SIZE 500K;

CREATE DSI NEWORDER_74_DSI

```
        DSO NEWORDER_DSO
          USING(1607,1628)
          ALLOCATE PRIME    ON SP147 SIZE 14352K
                            SP148 SIZE 14344K,
             OVERFLOW ON SP147 SIZE 501K
                            SP148 SIZE 500K;

CREATE DSI NEWORDER_75_DSI
        DSO NEWORDER_DSO
          USING(1629,1650)
          ALLOCATE PRIME    ON SP149 SIZE 14352K
                            SP150 SIZE 14344K,
             OVERFLOW ON SP149 SIZE 501K
                            SP150 SIZE 500K;

CREATE DSI NEWORDER_76_DSI
        DSO NEWORDER_DSO
          USING(1651,1672)
          ALLOCATE PRIME    ON SP151 SIZE 14352K
                            SP152 SIZE 14344K,
             OVERFLOW ON SP151 SIZE 501K
                            SP152 SIZE 500K;

CREATE DSI NEWORDER_77_DSI
        DSO NEWORDER_DSO
          USING(1673,1694)
          ALLOCATE PRIME    ON SP153 SIZE 14352K
                            SP154 SIZE 14344K,
             OVERFLOW ON SP153 SIZE 501K
                            SP154 SIZE 500K;

CREATE DSI NEWORDER_78_DSI
        DSO NEWORDER_DSO
          USING(1695,1716)
          ALLOCATE PRIME    ON SP155 SIZE 14352K
                            SP156 SIZE 14344K,
             OVERFLOW ON SP155 SIZE 501K
                            SP156 SIZE 500K;

CREATE DSI NEWORDER_79_DSI
        DSO NEWORDER_DSO
          USING(1717,1738)
          ALLOCATE PRIME    ON SP157 SIZE 14352K
                            SP158 SIZE 14344K,
             OVERFLOW ON SP157 SIZE 501K
                            SP158 SIZE 500K;

CREATE DSI NEWORDER_80_DSI
        DSO NEWORDER_DSO
          USING(1739,1760)
          ALLOCATE PRIME    ON SP159 SIZE 14352K
                            SP160 SIZE 14344K,
             OVERFLOW ON SP159 SIZE 501K
                            SP160 SIZE 500K;

CREATE DSI NEWORDER_81_DSI
        DSO NEWORDER_DSO
          USING(1761,1782)
          ALLOCATE PRIME    ON SP161 SIZE 14352K
                            SP162 SIZE 14344K,
             OVERFLOW ON SP161 SIZE 501K
                            SP162 SIZE 500K;

CREATE DSI NEWORDER_82_DSI
        DSO NEWORDER_DSO
          USING(1783,1804)
          ALLOCATE PRIME    ON SP163 SIZE 14352K
                            SP164 SIZE 14344K,
             OVERFLOW ON SP163 SIZE 501K
                            SP164 SIZE 500K;

CREATE DSI NEWORDER_83_DSI
        DSO NEWORDER_DSO
          USING(1805,1826)
          ALLOCATE PRIME    ON SP165 SIZE 14352K
                            SP166 SIZE 14344K,
             OVERFLOW ON SP165 SIZE 501K
                            SP166 SIZE 500K;

CREATE DSI NEWORDER_84_DSI
        DSO NEWORDER_DSO
          USING(1827,1848)
          ALLOCATE PRIME    ON SP167 SIZE 14352K
                            SP168 SIZE 14344K,
             OVERFLOW ON SP167 SIZE 501K
                            SP168 SIZE 500K;

CREATE DSI NEWORDER_85_DSI
        DSO NEWORDER_DSO
          USING(1849,1870)
          ALLOCATE PRIME    ON SP169 SIZE 14352K
                            SP170 SIZE 14344K,
             OVERFLOW ON SP169 SIZE 501K
                            SP170 SIZE 500K;

CREATE DSI NEWORDER_86_DSI
        DSO NEWORDER_DSO
          USING(1871,1892)
          ALLOCATE PRIME    ON SP171 SIZE 14352K
                            SP172 SIZE 14344K,
             OVERFLOW ON SP171 SIZE 501K
                            SP172 SIZE 500K;

CREATE DSI NEWORDER_87_DSI
        DSO NEWORDER_DSO
          USING(1893,1914)
          ALLOCATE PRIME    ON SP173 SIZE 14352K
                            SP174 SIZE 14344K,
             OVERFLOW ON SP173 SIZE 501K
                            SP174 SIZE 500K;

CREATE DSI NEWORDER_88_DSI
        DSO NEWORDER_DSO
          USING(1915,1936)
          ALLOCATE PRIME    ON SP175 SIZE 14352K
                            SP176 SIZE 14344K,
             OVERFLOW ON SP175 SIZE 501K
                            SP176 SIZE 500K;

CREATE DSI NEWORDER_89_DSI
        DSO NEWORDER_DSO
          USING(1937,1958)
          ALLOCATE PRIME    ON SP177 SIZE 14352K
                            SP178 SIZE 14344K,
             OVERFLOW ON SP177 SIZE 501K
                            SP178 SIZE 500K;
```

CREATE DSI NEWORDER_90_DSI
    DSO NEWORDER_DSO
        USING(1959,1980)
        ALLOCATE PRIME   ON SP179 SIZE 14352K
                              SP180 SIZE 14344K,
            OVERFLOW ON SP179 SIZE 501K
                              SP180 SIZE 500K;

CREATE DSI NEWORDER_91_DSI
    DSO NEWORDER_DSO
        USING(1981,2002)
        ALLOCATE PRIME   ON SP181 SIZE 14352K
                              SP182 SIZE 14344K,
            OVERFLOW ON SP181 SIZE 501K
                              SP182 SIZE 500K;

CREATE DSI NEWORDER_92_DSI
    DSO NEWORDER_DSO
        USING(2003,2024)
        ALLOCATE PRIME   ON SP183 SIZE 14352K
                              SP184 SIZE 14344K,
            OVERFLOW ON SP183 SIZE 501K
                              SP184 SIZE 500K;

CREATE DSI NEWORDER_93_DSI
    DSO NEWORDER_DSO
        USING(2025,2046)
        ALLOCATE PRIME   ON SP185 SIZE 14352K
                              SP186 SIZE 14344K,
            OVERFLOW ON SP185 SIZE 501K
                              SP186 SIZE 500K;

CREATE DSI NEWORDER_94_DSI
    DSO NEWORDER_DSO
        USING(2047,2068)
        ALLOCATE PRIME   ON SP187 SIZE 14352K
                              SP188 SIZE 14344K,
            OVERFLOW ON SP187 SIZE 501K
                              SP188 SIZE 500K;

CREATE DSI NEWORDER_95_DSI
    DSO NEWORDER_DSO
        USING(2069,2090)
        ALLOCATE PRIME   ON SP189 SIZE 14352K
                              SP190 SIZE 14344K,
            OVERFLOW ON SP189 SIZE 501K
                              SP190 SIZE 500K;

CREATE DSI NEWORDER_96_DSI
    DSO NEWORDER_DSO
        USING(2091,4224)
        ALLOCATE PRIME   ON SP191 SIZE 14352K
                              SP192 SIZE 14344K,
            OVERFLOW ON SP191 SIZE 501K
                              SP192 SIZE 500K;

::::::::::::
ddl.dat.OI
::::::::::::

-- * Phase.2-4b: Orders-IX

CREATE DSO ORDERS_IX_DSO
    INDEX ON TPCC_SCHEMA.ORDERS(O_W_ID,O_D_ID,O_C_ID)
    TYPE BTREE(PAGESIZE1(8),PAGESIZE2(32));

CREATE DSI ORDERS_IX_1_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_1_DSI
    ALLOCATE INDEX   ON SP1 SIZE 1024K,
        BASE   ON SP1 SIZE 16400K
                              SP2 SIZE 16392K;

CREATE DSI ORDERS_IX_2_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_2_DSI
    ALLOCATE INDEX   ON SP3 SIZE 1024K,
        BASE   ON SP3 SIZE 16400K
                              SP4 SIZE 16392K;

CREATE DSI ORDERS_IX_3_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_3_DSI
    ALLOCATE INDEX   ON SP5 SIZE 1024K,
        BASE   ON SP5 SIZE 16400K
                              SP6 SIZE 16392K;

CREATE DSI ORDERS_IX_4_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_4_DSI
    ALLOCATE INDEX   ON SP7 SIZE 1024K,
        BASE   ON SP7 SIZE 16400K
                              SP8 SIZE 16392K;

CREATE DSI ORDERS_IX_5_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_5_DSI
    ALLOCATE INDEX   ON SP9 SIZE 1024K,
        BASE   ON SP9 SIZE 16400K
                              SP10 SIZE 16392K;

CREATE DSI ORDERS_IX_6_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_6_DSI
    ALLOCATE INDEX   ON SP11 SIZE 1024K,
        BASE   ON SP11 SIZE 16400K
                              SP12 SIZE 16392K;

CREATE DSI ORDERS_IX_7_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_7_DSI
    ALLOCATE INDEX   ON SP13 SIZE 1024K,
        BASE   ON SP13 SIZE 16400K
                              SP14 SIZE 16392K;

CREATE DSI ORDERS_IX_8_DSI
    INDEX
    DSO ORDERS_IX_DSO

```
        BASE ORDERS_8_DSI
        ALLOCATE INDEX    ON SP15 SIZE 1024K,
            BASE     ON SP15 SIZE 16400K
                             SP16 SIZE 16392K;

CREATE DSI ORDERS_IX_9_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_9_DSI
        ALLOCATE INDEX    ON SP17 SIZE 1024K,
            BASE     ON SP17 SIZE 16400K
                             SP18 SIZE 16392K;

CREATE DSI ORDERS_IX_10_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_10_DSI
        ALLOCATE INDEX    ON SP19 SIZE 1024K,
            BASE     ON SP19 SIZE 16400K
                             SP20 SIZE 16392K;

CREATE DSI ORDERS_IX_11_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_11_DSI
        ALLOCATE INDEX    ON SP21 SIZE 1024K,
            BASE     ON SP21 SIZE 16400K
                             SP22 SIZE 16392K;

CREATE DSI ORDERS_IX_12_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_12_DSI
        ALLOCATE INDEX    ON SP23 SIZE 1024K,
            BASE     ON SP23 SIZE 16400K
                             SP24 SIZE 16392K;

CREATE DSI ORDERS_IX_13_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_13_DSI
        ALLOCATE INDEX    ON SP25 SIZE 1024K,
            BASE     ON SP25 SIZE 16400K
                             SP26 SIZE 16392K;

CREATE DSI ORDERS_IX_14_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_14_DSI
        ALLOCATE INDEX    ON SP27 SIZE 1024K,
            BASE     ON SP27 SIZE 16400K
                             SP28 SIZE 16392K;

CREATE DSI ORDERS_IX_15_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_15_DSI
        ALLOCATE INDEX    ON SP29 SIZE 1024K,
            BASE     ON SP29 SIZE 16400K
                             SP30 SIZE 16392K;

CREATE DSI ORDERS_IX_16_DSI
        INDEX


        DSO ORDERS_IX_DSO
        BASE ORDERS_16_DSI
        ALLOCATE INDEX    ON SP31 SIZE 1024K,
            BASE     ON SP31 SIZE 16400K
                             SP32 SIZE 16392K;

CREATE DSI ORDERS_IX_17_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_17_DSI
        ALLOCATE INDEX    ON SP33 SIZE 1024K,
            BASE     ON SP33 SIZE 16400K
                             SP34 SIZE 16392K;

CREATE DSI ORDERS_IX_18_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_18_DSI
        ALLOCATE INDEX    ON SP35 SIZE 1024K,
            BASE     ON SP35 SIZE 16400K
                             SP36 SIZE 16392K;

CREATE DSI ORDERS_IX_19_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_19_DSI
        ALLOCATE INDEX    ON SP37 SIZE 1024K,
            BASE     ON SP37 SIZE 16400K
                             SP38 SIZE 16392K;

CREATE DSI ORDERS_IX_20_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_20_DSI
        ALLOCATE INDEX    ON SP39 SIZE 1024K,
            BASE     ON SP39 SIZE 16400K
                             SP40 SIZE 16392K;

CREATE DSI ORDERS_IX_21_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_21_DSI
        ALLOCATE INDEX    ON SP41 SIZE 1024K,
            BASE     ON SP41 SIZE 16400K
                             SP42 SIZE 16392K;

CREATE DSI ORDERS_IX_22_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_22_DSI
        ALLOCATE INDEX    ON SP43 SIZE 1024K,
            BASE     ON SP43 SIZE 16400K
                             SP44 SIZE 16392K;

CREATE DSI ORDERS_IX_23_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_23_DSI
        ALLOCATE INDEX    ON SP45 SIZE 1024K,
            BASE     ON SP45 SIZE 16400K
                             SP46 SIZE 16392K;

CREATE DSI ORDERS_IX_24_DSI
```

```
                INDEX                                    CREATE DSI ORDERS_IX_32_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_24_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP47 SIZE 1024K,           BASE ORDERS_32_DSI
                       BASE     ON SP47 SIZE 16400K             ALLOCATE INDEX    ON SP63 SIZE 1024K,
                                       SP48 SIZE 16392K;               BASE     ON SP63 SIZE 16400K
                                                                                SP64 SIZE 16392K;
        CREATE DSI ORDERS_IX_25_DSI
                INDEX                                    CREATE DSI ORDERS_IX_33_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_25_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP49 SIZE 1024K,           BASE ORDERS_33_DSI
                       BASE     ON SP49 SIZE 16400K             ALLOCATE INDEX    ON SP65 SIZE 1024K,
                                       SP50 SIZE 16392K;               BASE     ON SP65 SIZE 16400K
                                                                                SP66 SIZE 16392K;
        CREATE DSI ORDERS_IX_26_DSI
                INDEX                                    CREATE DSI ORDERS_IX_34_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_26_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP51 SIZE 1024K,           BASE ORDERS_34_DSI
                       BASE     ON SP51 SIZE 16400K             ALLOCATE INDEX    ON SP67 SIZE 1024K,
                                       SP52 SIZE 16392K;               BASE     ON SP67 SIZE 16400K
                                                                                SP68 SIZE 16392K;
        CREATE DSI ORDERS_IX_27_DSI
                INDEX                                    CREATE DSI ORDERS_IX_35_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_27_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP53 SIZE 1024K,           BASE ORDERS_35_DSI
                       BASE     ON SP53 SIZE 16400K             ALLOCATE INDEX    ON SP69 SIZE 1024K,
                                       SP54 SIZE 16392K;               BASE     ON SP69 SIZE 16400K
                                                                                SP70 SIZE 16392K;
        CREATE DSI ORDERS_IX_28_DSI
                INDEX                                    CREATE DSI ORDERS_IX_36_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_28_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP55 SIZE 1024K,           BASE ORDERS_36_DSI
                       BASE     ON SP55 SIZE 16400K             ALLOCATE INDEX    ON SP71 SIZE 1024K,
                                       SP56 SIZE 16392K;               BASE     ON SP71 SIZE 16400K
                                                                                SP72 SIZE 16392K;
        CREATE DSI ORDERS_IX_29_DSI
                INDEX                                    CREATE DSI ORDERS_IX_37_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_29_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP57 SIZE 1024K,           BASE ORDERS_37_DSI
                       BASE     ON SP57 SIZE 16400K             ALLOCATE INDEX    ON SP73 SIZE 1024K,
                                       SP58 SIZE 16392K;               BASE     ON SP73 SIZE 16400K
                                                                                SP74 SIZE 16392K;
        CREATE DSI ORDERS_IX_30_DSI
                INDEX                                    CREATE DSI ORDERS_IX_38_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_30_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP59 SIZE 1024K,           BASE ORDERS_38_DSI
                       BASE     ON SP59 SIZE 16400K             ALLOCATE INDEX    ON SP75 SIZE 1024K,
                                       SP60 SIZE 16392K;               BASE     ON SP75 SIZE 16400K
                                                                                SP76 SIZE 16392K;
        CREATE DSI ORDERS_IX_31_DSI
                INDEX                                    CREATE DSI ORDERS_IX_39_DSI
                DSO ORDERS_IX_DSO                                INDEX
                BASE ORDERS_31_DSI                              DSO ORDERS_IX_DSO
                ALLOCATE INDEX    ON SP61 SIZE 1024K,           BASE ORDERS_39_DSI
                       BASE     ON SP61 SIZE 16400K             ALLOCATE INDEX    ON SP77 SIZE 1024K,
                                       SP62 SIZE 16392K;               BASE     ON SP77 SIZE 16400K
                                                                                SP78 SIZE 16392K;
```

SP94 SIZE 16392K;

```
CREATE DSI ORDERS_IX_40_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_40_DSI
     ALLOCATE INDEX    ON SP79 SIZE 1024K,
          BASE    ON SP79 SIZE 16400K
                              SP80 SIZE 16392K;

CREATE DSI ORDERS_IX_41_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_41_DSI
     ALLOCATE INDEX    ON SP81 SIZE 1024K,
          BASE    ON SP81 SIZE 16400K
                              SP82 SIZE 16392K;

CREATE DSI ORDERS_IX_42_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_42_DSI
     ALLOCATE INDEX    ON SP83 SIZE 1024K,
          BASE    ON SP83 SIZE 16400K
                              SP84 SIZE 16392K;

CREATE DSI ORDERS_IX_43_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_43_DSI
     ALLOCATE INDEX    ON SP85 SIZE 1024K,
          BASE    ON SP85 SIZE 16400K
                              SP86 SIZE 16392K;

CREATE DSI ORDERS_IX_44_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_44_DSI
     ALLOCATE INDEX    ON SP87 SIZE 1024K,
          BASE    ON SP87 SIZE 16400K
                              SP88 SIZE 16392K;

CREATE DSI ORDERS_IX_45_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_45_DSI
     ALLOCATE INDEX    ON SP89 SIZE 1024K,
          BASE    ON SP89 SIZE 16400K
                              SP90 SIZE 16392K;

CREATE DSI ORDERS_IX_46_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_46_DSI
     ALLOCATE INDEX    ON SP91 SIZE 1024K,
          BASE    ON SP91 SIZE 16400K
                              SP92 SIZE 16392K;

CREATE DSI ORDERS_IX_47_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_47_DSI
     ALLOCATE INDEX    ON SP93 SIZE 1024K,
          BASE    ON SP93 SIZE 16400K
```

```
CREATE DSI ORDERS_IX_48_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_48_DSI
     ALLOCATE INDEX    ON SP95 SIZE 1024K,
          BASE    ON SP95 SIZE 16400K
                              SP96 SIZE 16392K;

CREATE DSI ORDERS_IX_49_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_49_DSI
     ALLOCATE INDEX    ON SP97 SIZE 1024K,
          BASE    ON SP97 SIZE 16400K
                              SP98 SIZE 16392K;

CREATE DSI ORDERS_IX_50_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_50_DSI
     ALLOCATE INDEX    ON SP99 SIZE 1024K,
          BASE    ON SP99 SIZE 16400K
                              SP100 SIZE 16392K;

CREATE DSI ORDERS_IX_51_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_51_DSI
     ALLOCATE INDEX    ON SP101 SIZE 1024K,
          BASE    ON SP101 SIZE 16400K
                              SP102 SIZE 16392K;

CREATE DSI ORDERS_IX_52_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_52_DSI
     ALLOCATE INDEX    ON SP103 SIZE 1024K,
          BASE    ON SP103 SIZE 16400K
                              SP104 SIZE 16392K;

CREATE DSI ORDERS_IX_53_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_53_DSI
     ALLOCATE INDEX    ON SP105 SIZE 1024K,
          BASE    ON SP105 SIZE 16400K
                              SP106 SIZE 16392K;

CREATE DSI ORDERS_IX_54_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_54_DSI
     ALLOCATE INDEX    ON SP107 SIZE 1024K,
          BASE    ON SP107 SIZE 16400K
                              SP108 SIZE 16392K;

CREATE DSI ORDERS_IX_55_DSI
     INDEX
     DSO ORDERS_IX_DSO
     BASE ORDERS_55_DSI
     ALLOCATE INDEX    ON SP109 SIZE 1024K,
```

BASE   ON SP109 SIZE 16400K
SP110 SIZE 16392K;

CREATE DSI ORDERS_IX_56_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_56_DSI
    ALLOCATE INDEX   ON SP111 SIZE 1024K,
        BASE   ON SP111 SIZE 16400K
SP112 SIZE 16392K;

CREATE DSI ORDERS_IX_57_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_57_DSI
    ALLOCATE INDEX   ON SP113 SIZE 1024K,
        BASE   ON SP113 SIZE 16400K
SP114 SIZE 16392K;

CREATE DSI ORDERS_IX_58_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_58_DSI
    ALLOCATE INDEX   ON SP115 SIZE 1024K,
        BASE   ON SP115 SIZE 16400K
SP116 SIZE 16392K;

CREATE DSI ORDERS_IX_59_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_59_DSI
    ALLOCATE INDEX   ON SP117 SIZE 1024K,
        BASE   ON SP117 SIZE 16400K
SP118 SIZE 16392K;

CREATE DSI ORDERS_IX_60_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_60_DSI
    ALLOCATE INDEX   ON SP119 SIZE 1024K,
        BASE   ON SP119 SIZE 16400K
SP120 SIZE 16392K;

CREATE DSI ORDERS_IX_61_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_61_DSI
    ALLOCATE INDEX   ON SP121 SIZE 1024K,
        BASE   ON SP121 SIZE 16400K
SP122 SIZE 16392K;

CREATE DSI ORDERS_IX_62_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_62_DSI
    ALLOCATE INDEX   ON SP123 SIZE 1024K,
        BASE   ON SP123 SIZE 16400K
SP124 SIZE 16392K;

CREATE DSI ORDERS_IX_63_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_63_DSI

ALLOCATE INDEX   ON SP125 SIZE 1024K,
        BASE   ON SP125 SIZE 16400K
SP126 SIZE 16392K;

CREATE DSI ORDERS_IX_64_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_64_DSI
    ALLOCATE INDEX   ON SP127 SIZE 1024K,
        BASE   ON SP127 SIZE 16400K
SP128 SIZE 16392K;

CREATE DSI ORDERS_IX_65_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_65_DSI
    ALLOCATE INDEX   ON SP129 SIZE 1024K,
        BASE   ON SP129 SIZE 16400K
SP130 SIZE 16392K;

CREATE DSI ORDERS_IX_66_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_66_DSI
    ALLOCATE INDEX   ON SP131 SIZE 1024K,
        BASE   ON SP131 SIZE 16400K
SP132 SIZE 16392K;

CREATE DSI ORDERS_IX_67_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_67_DSI
    ALLOCATE INDEX   ON SP133 SIZE 1024K,
        BASE   ON SP133 SIZE 16400K
SP134 SIZE 16392K;

CREATE DSI ORDERS_IX_68_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_68_DSI
    ALLOCATE INDEX   ON SP135 SIZE 1024K,
        BASE   ON SP135 SIZE 16400K
SP136 SIZE 16392K;

CREATE DSI ORDERS_IX_69_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_69_DSI
    ALLOCATE INDEX   ON SP137 SIZE 1024K,
        BASE   ON SP137 SIZE 16400K
SP138 SIZE 16392K;

CREATE DSI ORDERS_IX_70_DSI
    INDEX
    DSO ORDERS_IX_DSO
    BASE ORDERS_70_DSI
    ALLOCATE INDEX   ON SP139 SIZE 1024K,
        BASE   ON SP139 SIZE 16400K
SP140 SIZE 16392K;

CREATE DSI ORDERS_IX_71_DSI
    INDEX
    DSO ORDERS_IX_DSO

```
        BASE ORDERS_71_DSI
        ALLOCATE INDEX    ON SP141 SIZE 1024K,
            BASE     ON SP141 SIZE 16400K
                              SP142 SIZE 16392K;

CREATE DSI ORDERS_IX_72_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_72_DSI
        ALLOCATE INDEX    ON SP143 SIZE 1024K,
            BASE     ON SP143 SIZE 16400K
                              SP144 SIZE 16392K;

CREATE DSI ORDERS_IX_73_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_73_DSI
        ALLOCATE INDEX    ON SP145 SIZE 1024K,
            BASE     ON SP145 SIZE 16400K
                              SP146 SIZE 16392K;

CREATE DSI ORDERS_IX_74_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_74_DSI
        ALLOCATE INDEX    ON SP147 SIZE 1024K,
            BASE     ON SP147 SIZE 16400K
                              SP148 SIZE 16392K;

CREATE DSI ORDERS_IX_75_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_75_DSI
        ALLOCATE INDEX    ON SP149 SIZE 1024K,
            BASE     ON SP149 SIZE 16400K
                              SP150 SIZE 16392K;

CREATE DSI ORDERS_IX_76_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_76_DSI
        ALLOCATE INDEX    ON SP151 SIZE 1024K,
            BASE     ON SP151 SIZE 16400K
                              SP152 SIZE 16392K;

CREATE DSI ORDERS_IX_77_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_77_DSI
        ALLOCATE INDEX    ON SP153 SIZE 1024K,
            BASE     ON SP153 SIZE 16400K
                              SP154 SIZE 16392K;

CREATE DSI ORDERS_IX_78_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_78_DSI
        ALLOCATE INDEX    ON SP155 SIZE 1024K,
            BASE     ON SP155 SIZE 16400K
                              SP156 SIZE 16392K;

CREATE DSI ORDERS_IX_79_DSI
        INDEX
```

```
        DSO ORDERS_IX_DSO
        BASE ORDERS_79_DSI
        ALLOCATE INDEX    ON SP157 SIZE 1024K,
            BASE     ON SP157 SIZE 16400K
                              SP158 SIZE 16392K;

CREATE DSI ORDERS_IX_80_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_80_DSI
        ALLOCATE INDEX    ON SP159 SIZE 1024K,
            BASE     ON SP159 SIZE 16400K
                              SP160 SIZE 16392K;

CREATE DSI ORDERS_IX_81_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_81_DSI
        ALLOCATE INDEX    ON SP161 SIZE 1024K,
            BASE     ON SP161 SIZE 16400K
                              SP162 SIZE 16392K;

CREATE DSI ORDERS_IX_82_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_82_DSI
        ALLOCATE INDEX    ON SP163 SIZE 1024K,
            BASE     ON SP163 SIZE 16400K
                              SP164 SIZE 16392K;

CREATE DSI ORDERS_IX_83_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_83_DSI
        ALLOCATE INDEX    ON SP165 SIZE 1024K,
            BASE     ON SP165 SIZE 16400K
                              SP166 SIZE 16392K;

CREATE DSI ORDERS_IX_84_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_84_DSI
        ALLOCATE INDEX    ON SP167 SIZE 1024K,
            BASE     ON SP167 SIZE 16400K
                              SP168 SIZE 16392K;

CREATE DSI ORDERS_IX_85_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_85_DSI
        ALLOCATE INDEX    ON SP169 SIZE 1024K,
            BASE     ON SP169 SIZE 16400K
                              SP170 SIZE 16392K;

CREATE DSI ORDERS_IX_86_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_86_DSI
        ALLOCATE INDEX    ON SP171 SIZE 1024K,
            BASE     ON SP171 SIZE 16400K
                              SP172 SIZE 16392K;

CREATE DSI ORDERS_IX_87_DSI
```

```
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_87_DSI
        ALLOCATE INDEX    ON SP173 SIZE 1024K,
            BASE     ON SP173 SIZE 16400K
                            SP174 SIZE 16392K;

CREATE DSI ORDERS_IX_88_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_88_DSI
        ALLOCATE INDEX    ON SP175 SIZE 1024K,
            BASE     ON SP175 SIZE 16400K
                            SP176 SIZE 16392K;

CREATE DSI ORDERS_IX_89_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_89_DSI
        ALLOCATE INDEX    ON SP177 SIZE 1024K,
            BASE     ON SP177 SIZE 16400K
                            SP178 SIZE 16392K;

CREATE DSI ORDERS_IX_90_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_90_DSI
        ALLOCATE INDEX    ON SP179 SIZE 1024K,
            BASE     ON SP179 SIZE 16400K
                            SP180 SIZE 16392K;

CREATE DSI ORDERS_IX_91_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_91_DSI
        ALLOCATE INDEX    ON SP181 SIZE 1024K,
            BASE     ON SP181 SIZE 16400K
                            SP182 SIZE 16392K;

CREATE DSI ORDERS_IX_92_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_92_DSI
        ALLOCATE INDEX    ON SP183 SIZE 1024K,
            BASE     ON SP183 SIZE 16400K
                            SP184 SIZE 16392K;

CREATE DSI ORDERS_IX_93_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_93_DSI
        ALLOCATE INDEX    ON SP185 SIZE 1024K,
            BASE     ON SP185 SIZE 16400K
                            SP186 SIZE 16392K;

CREATE DSI ORDERS_IX_94_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_94_DSI
        ALLOCATE INDEX    ON SP187 SIZE 1024K,
            BASE     ON SP187 SIZE 16400K
                            SP188 SIZE 16392K;

CREATE DSI ORDERS_IX_95_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_95_DSI
        ALLOCATE INDEX    ON SP189 SIZE 1024K,
            BASE     ON SP189 SIZE 16400K
                            SP190 SIZE 16392K;

CREATE DSI ORDERS_IX_96_DSI
        INDEX
        DSO ORDERS_IX_DSO
        BASE ORDERS_96_DSI
        ALLOCATE INDEX    ON SP191 SIZE 1024K,
            BASE     ON SP191 SIZE 16400K
                            SP192 SIZE 16392K;

::::::::::::::
ddl.dat.OL
::::::::::::::
--------------------------------------------------------------------------
-- * Phase.2-5a: OrderLine
--------------------------------------------------------------------------
CREATE DSO ORDERLINE_DSO
        FROM TPCC_SCHEMA.ORDERLINE
            TYPE
RANDOM(PAGESIZE1(32),PAGESIZE2(8),RULE((OL_O_ID∕30)*110+OL_W_ID*10
+OL_D_ID+(OL_NUMBER+(OL_O_ID-((OL_O_ID∕30)*30))*15)*13757))
            WHERE (OL_W_ID) BETWEEN (?) AND (?);

CREATE DSI ORDERLIN_1_DSI
        DSO ORDERLINE_DSO
        USING(1,11)
        ALLOCATE PRIME    ON SP1 SIZE 440256K,
            OVERFLOW ON SP1 SIZE  1760K;

CREATE DSI ORDERLIN_2_DSI
        DSO ORDERLINE_DSO
        USING(12,22)
        ALLOCATE PRIME    ON SP2 SIZE 440256K,
            OVERFLOW ON SP2 SIZE  1760K;

CREATE DSI ORDERLIN_3_DSI
        DSO ORDERLINE_DSO
        USING(23,33)
        ALLOCATE PRIME    ON SP3 SIZE 440256K,
            OVERFLOW ON SP3 SIZE  1760K;

CREATE DSI ORDERLIN_4_DSI
        DSO ORDERLINE_DSO
        USING(34,44)
        ALLOCATE PRIME    ON SP4 SIZE 440256K,
            OVERFLOW ON SP4 SIZE  1760K;

CREATE DSI ORDERLIN_5_DSI
        DSO ORDERLINE_DSO
        USING(45,55)
        ALLOCATE PRIME    ON SP5 SIZE 440256K,
            OVERFLOW ON SP5 SIZE  1760K;

CREATE DSI ORDERLIN_6_DSI
        DSO ORDERLINE_DSO
        USING(56,66)
        ALLOCATE PRIME    ON SP6 SIZE 440256K,
```

OVERFLOW ON SP6 SIZE 1760K;

CREATE DSI ORDERLIN_7_DSI
    DSO ORDERLINE_DSO
     USING(67,77)
     ALLOCATE PRIME   ON SP7 SIZE 440256K,
        OVERFLOW ON SP7 SIZE 1760K;

CREATE DSI ORDERLIN_8_DSI
    DSO ORDERLINE_DSO
     USING(78,88)
     ALLOCATE PRIME   ON SP8 SIZE 440256K,
        OVERFLOW ON SP8 SIZE 1760K;

CREATE DSI ORDERLIN_9_DSI
    DSO ORDERLINE_DSO
     USING(89,99)
     ALLOCATE PRIME   ON SP9 SIZE 440256K,
        OVERFLOW ON SP9 SIZE 1760K;

CREATE DSI ORDERLIN_10_DSI
    DSO ORDERLINE_DSO
     USING(100,110)
     ALLOCATE PRIME   ON SP10 SIZE 440256K,
        OVERFLOW ON SP10 SIZE 1760K;

CREATE DSI ORDERLIN_11_DSI
    DSO ORDERLINE_DSO
     USING(111,121)
     ALLOCATE PRIME   ON SP11 SIZE 440256K,
        OVERFLOW ON SP11 SIZE 1760K;

CREATE DSI ORDERLIN_12_DSI
    DSO ORDERLINE_DSO
     USING(122,132)
     ALLOCATE PRIME   ON SP12 SIZE 440256K,
        OVERFLOW ON SP12 SIZE 1760K;

CREATE DSI ORDERLIN_13_DSI
    DSO ORDERLINE_DSO
     USING(133,143)
     ALLOCATE PRIME   ON SP13 SIZE 440256K,
        OVERFLOW ON SP13 SIZE 1760K;

CREATE DSI ORDERLIN_14_DSI
    DSO ORDERLINE_DSO
     USING(144,154)
     ALLOCATE PRIME   ON SP14 SIZE 440256K,
        OVERFLOW ON SP14 SIZE 1760K;

CREATE DSI ORDERLIN_15_DSI
    DSO ORDERLINE_DSO
     USING(155,165)
     ALLOCATE PRIME   ON SP15 SIZE 440256K,
        OVERFLOW ON SP15 SIZE 1760K;

CREATE DSI ORDERLIN_16_DSI
    DSO ORDERLINE_DSO
     USING(166,176)
     ALLOCATE PRIME   ON SP16 SIZE 440256K,
        OVERFLOW ON SP16 SIZE 1760K;

CREATE DSI ORDERLIN_17_DSI

DSO ORDERLINE_DSO
     USING(177,187)
     ALLOCATE PRIME   ON SP17 SIZE 440256K,
        OVERFLOW ON SP17 SIZE 1760K;

CREATE DSI ORDERLIN_18_DSI
    DSO ORDERLINE_DSO
     USING(188,198)
     ALLOCATE PRIME   ON SP18 SIZE 440256K,
        OVERFLOW ON SP18 SIZE 1760K;

CREATE DSI ORDERLIN_19_DSI
    DSO ORDERLINE_DSO
     USING(199,209)
     ALLOCATE PRIME   ON SP19 SIZE 440256K,
        OVERFLOW ON SP19 SIZE 1760K;

CREATE DSI ORDERLIN_20_DSI
    DSO ORDERLINE_DSO
     USING(210,220)
     ALLOCATE PRIME   ON SP20 SIZE 440256K,
        OVERFLOW ON SP20 SIZE 1760K;

CREATE DSI ORDERLIN_21_DSI
    DSO ORDERLINE_DSO
     USING(221,231)
     ALLOCATE PRIME   ON SP21 SIZE 440256K,
        OVERFLOW ON SP21 SIZE 1760K;

CREATE DSI ORDERLIN_22_DSI
    DSO ORDERLINE_DSO
     USING(232,242)
     ALLOCATE PRIME   ON SP22 SIZE 440256K,
        OVERFLOW ON SP22 SIZE 1760K;

CREATE DSI ORDERLIN_23_DSI
    DSO ORDERLINE_DSO
     USING(243,253)
     ALLOCATE PRIME   ON SP23 SIZE 440256K,
        OVERFLOW ON SP23 SIZE 1760K;

CREATE DSI ORDERLIN_24_DSI
    DSO ORDERLINE_DSO
     USING(254,264)
     ALLOCATE PRIME   ON SP24 SIZE 440256K,
        OVERFLOW ON SP24 SIZE 1760K;

CREATE DSI ORDERLIN_25_DSI
    DSO ORDERLINE_DSO
     USING(265,275)
     ALLOCATE PRIME   ON SP25 SIZE 440256K,
        OVERFLOW ON SP25 SIZE 1760K;

CREATE DSI ORDERLIN_26_DSI
    DSO ORDERLINE_DSO
     USING(276,286)
     ALLOCATE PRIME   ON SP26 SIZE 440256K,
        OVERFLOW ON SP26 SIZE 1760K;

CREATE DSI ORDERLIN_27_DSI
    DSO ORDERLINE_DSO
     USING(287,297)
     ALLOCATE PRIME   ON SP27 SIZE 440256K,

OVERFLOW ON SP27 SIZE 1760K;

CREATE DSI ORDERLIN_28_DSI
    DSO ORDERLINE_DSO
     USING(298,308)
      ALLOCATE PRIME   ON SP28 SIZE 440256K,
          OVERFLOW ON SP28 SIZE 1760K;

CREATE DSI ORDERLIN_29_DSI
    DSO ORDERLINE_DSO
     USING(309,319)
      ALLOCATE PRIME   ON SP29 SIZE 440256K,
          OVERFLOW ON SP29 SIZE 1760K;

CREATE DSI ORDERLIN_30_DSI
    DSO ORDERLINE_DSO
     USING(320,330)
      ALLOCATE PRIME   ON SP30 SIZE 440256K,
          OVERFLOW ON SP30 SIZE 1760K;

CREATE DSI ORDERLIN_31_DSI
    DSO ORDERLINE_DSO
     USING(331,341)
      ALLOCATE PRIME   ON SP31 SIZE 440256K,
          OVERFLOW ON SP31 SIZE 1760K;

CREATE DSI ORDERLIN_32_DSI
    DSO ORDERLINE_DSO
     USING(342,352)
      ALLOCATE PRIME   ON SP32 SIZE 440256K,
          OVERFLOW ON SP32 SIZE 1760K;

CREATE DSI ORDERLIN_33_DSI
    DSO ORDERLINE_DSO
     USING(353,363)
      ALLOCATE PRIME   ON SP33 SIZE 440256K,
          OVERFLOW ON SP33 SIZE 1760K;

CREATE DSI ORDERLIN_34_DSI
    DSO ORDERLINE_DSO
     USING(364,374)
      ALLOCATE PRIME   ON SP34 SIZE 440256K,
          OVERFLOW ON SP34 SIZE 1760K;

CREATE DSI ORDERLIN_35_DSI
    DSO ORDERLINE_DSO
     USING(375,385)
      ALLOCATE PRIME   ON SP35 SIZE 440256K,
          OVERFLOW ON SP35 SIZE 1760K;

CREATE DSI ORDERLIN_36_DSI
    DSO ORDERLINE_DSO
     USING(386,396)
      ALLOCATE PRIME   ON SP36 SIZE 440256K,
          OVERFLOW ON SP36 SIZE 1760K;

CREATE DSI ORDERLIN_37_DSI
    DSO ORDERLINE_DSO
     USING(397,407)
      ALLOCATE PRIME   ON SP37 SIZE 440256K,
          OVERFLOW ON SP37 SIZE 1760K;

CREATE DSI ORDERLIN_38_DSI

DSO ORDERLINE_DSO
     USING(408,418)
      ALLOCATE PRIME   ON SP38 SIZE 440256K,
          OVERFLOW ON SP38 SIZE 1760K;

CREATE DSI ORDERLIN_39_DSI
    DSO ORDERLINE_DSO
     USING(419,429)
      ALLOCATE PRIME   ON SP39 SIZE 440256K,
          OVERFLOW ON SP39 SIZE 1760K;

CREATE DSI ORDERLIN_40_DSI
    DSO ORDERLINE_DSO
     USING(430,440)
      ALLOCATE PRIME   ON SP40 SIZE 440256K,
          OVERFLOW ON SP40 SIZE 1760K;

CREATE DSI ORDERLIN_41_DSI
    DSO ORDERLINE_DSO
     USING(441,451)
      ALLOCATE PRIME   ON SP41 SIZE 440256K,
          OVERFLOW ON SP41 SIZE 1760K;

CREATE DSI ORDERLIN_42_DSI
    DSO ORDERLINE_DSO
     USING(452,462)
      ALLOCATE PRIME   ON SP42 SIZE 440256K,
          OVERFLOW ON SP42 SIZE 1760K;

CREATE DSI ORDERLIN_43_DSI
    DSO ORDERLINE_DSO
     USING(463,473)
      ALLOCATE PRIME   ON SP43 SIZE 440256K,
          OVERFLOW ON SP43 SIZE 1760K;

CREATE DSI ORDERLIN_44_DSI
    DSO ORDERLINE_DSO
     USING(474,484)
      ALLOCATE PRIME   ON SP44 SIZE 440256K,
          OVERFLOW ON SP44 SIZE 1760K;

CREATE DSI ORDERLIN_45_DSI
    DSO ORDERLINE_DSO
     USING(485,495)
      ALLOCATE PRIME   ON SP45 SIZE 440256K,
          OVERFLOW ON SP45 SIZE 1760K;

CREATE DSI ORDERLIN_46_DSI
    DSO ORDERLINE_DSO
     USING(496,506)
      ALLOCATE PRIME   ON SP46 SIZE 440256K,
          OVERFLOW ON SP46 SIZE 1760K;

CREATE DSI ORDERLIN_47_DSI
    DSO ORDERLINE_DSO
     USING(507,517)
      ALLOCATE PRIME   ON SP47 SIZE 440256K,
          OVERFLOW ON SP47 SIZE 1760K;

CREATE DSI ORDERLIN_48_DSI
    DSO ORDERLINE_DSO
     USING(518,528)
      ALLOCATE PRIME   ON SP48 SIZE 440256K,

OVERFLOW ON SP48 SIZE 1760K;

CREATE DSI ORDERLIN_49_DSI
    DSO ORDERLINE_DSO
     USING(529,539)
     ALLOCATE PRIME   ON SP49 SIZE 440256K,
        OVERFLOW ON SP49 SIZE 1760K;

CREATE DSI ORDERLIN_50_DSI
    DSO ORDERLINE_DSO
     USING(540,550)
     ALLOCATE PRIME   ON SP50 SIZE 440256K,
        OVERFLOW ON SP50 SIZE 1760K;

CREATE DSI ORDERLIN_51_DSI
    DSO ORDERLINE_DSO
     USING(551,561)
     ALLOCATE PRIME   ON SP51 SIZE 440256K,
        OVERFLOW ON SP51 SIZE 1760K;

CREATE DSI ORDERLIN_52_DSI
    DSO ORDERLINE_DSO
     USING(562,572)
     ALLOCATE PRIME   ON SP52 SIZE 440256K,
        OVERFLOW ON SP52 SIZE 1760K;

CREATE DSI ORDERLIN_53_DSI
    DSO ORDERLINE_DSO
     USING(573,583)
     ALLOCATE PRIME   ON SP53 SIZE 440256K,
        OVERFLOW ON SP53 SIZE 1760K;

CREATE DSI ORDERLIN_54_DSI
    DSO ORDERLINE_DSO
     USING(584,594)
     ALLOCATE PRIME   ON SP54 SIZE 440256K,
        OVERFLOW ON SP54 SIZE 1760K;

CREATE DSI ORDERLIN_55_DSI
    DSO ORDERLINE_DSO
     USING(595,605)
     ALLOCATE PRIME   ON SP55 SIZE 440256K,
        OVERFLOW ON SP55 SIZE 1760K;

CREATE DSI ORDERLIN_56_DSI
    DSO ORDERLINE_DSO
     USING(606,616)
     ALLOCATE PRIME   ON SP56 SIZE 440256K,
        OVERFLOW ON SP56 SIZE 1760K;

CREATE DSI ORDERLIN_57_DSI
    DSO ORDERLINE_DSO
     USING(617,627)
     ALLOCATE PRIME   ON SP57 SIZE 440256K,
        OVERFLOW ON SP57 SIZE 1760K;

CREATE DSI ORDERLIN_58_DSI
    DSO ORDERLINE_DSO
     USING(628,638)
     ALLOCATE PRIME   ON SP58 SIZE 440256K,
        OVERFLOW ON SP58 SIZE 1760K;

CREATE DSI ORDERLIN_59_DSI

DSO ORDERLINE_DSO
     USING(639,649)
     ALLOCATE PRIME   ON SP59 SIZE 440256K,
        OVERFLOW ON SP59 SIZE 1760K;

CREATE DSI ORDERLIN_60_DSI
    DSO ORDERLINE_DSO
     USING(650,660)
     ALLOCATE PRIME   ON SP60 SIZE 440256K,
        OVERFLOW ON SP60 SIZE 1760K;

CREATE DSI ORDERLIN_61_DSI
    DSO ORDERLINE_DSO
     USING(661,671)
     ALLOCATE PRIME   ON SP61 SIZE 440256K,
        OVERFLOW ON SP61 SIZE 1760K;

CREATE DSI ORDERLIN_62_DSI
    DSO ORDERLINE_DSO
     USING(672,682)
     ALLOCATE PRIME   ON SP62 SIZE 440256K,
        OVERFLOW ON SP62 SIZE 1760K;

CREATE DSI ORDERLIN_63_DSI
    DSO ORDERLINE_DSO
     USING(683,693)
     ALLOCATE PRIME   ON SP63 SIZE 440256K,
        OVERFLOW ON SP63 SIZE 1760K;

CREATE DSI ORDERLIN_64_DSI
    DSO ORDERLINE_DSO
     USING(694,704)
     ALLOCATE PRIME   ON SP64 SIZE 440256K,
        OVERFLOW ON SP64 SIZE 1760K;

CREATE DSI ORDERLIN_65_DSI
    DSO ORDERLINE_DSO
     USING(705,715)
     ALLOCATE PRIME   ON SP65 SIZE 440256K,
        OVERFLOW ON SP65 SIZE 1760K;

CREATE DSI ORDERLIN_66_DSI
    DSO ORDERLINE_DSO
     USING(716,726)
     ALLOCATE PRIME   ON SP66 SIZE 440256K,
        OVERFLOW ON SP66 SIZE 1760K;

CREATE DSI ORDERLIN_67_DSI
    DSO ORDERLINE_DSO
     USING(727,737)
     ALLOCATE PRIME   ON SP67 SIZE 440256K,
        OVERFLOW ON SP67 SIZE 1760K;

CREATE DSI ORDERLIN_68_DSI
    DSO ORDERLINE_DSO
     USING(738,748)
     ALLOCATE PRIME   ON SP68 SIZE 440256K,
        OVERFLOW ON SP68 SIZE 1760K;

CREATE DSI ORDERLIN_69_DSI
    DSO ORDERLINE_DSO
     USING(749,759)
     ALLOCATE PRIME   ON SP69 SIZE 440256K,

OVERFLOW ON SP69 SIZE 1760K;

CREATE DSI ORDERLIN_70_DSI
     DSO ORDERLINE_DSO
       USING(760,770)
         ALLOCATE PRIME   ON SP70 SIZE 440256K,
               OVERFLOW ON SP70 SIZE 1760K;

CREATE DSI ORDERLIN_71_DSI
     DSO ORDERLINE_DSO
       USING(771,781)
         ALLOCATE PRIME   ON SP71 SIZE 440256K,
               OVERFLOW ON SP71 SIZE 1760K;

CREATE DSI ORDERLIN_72_DSI
     DSO ORDERLINE_DSO
       USING(782,792)
         ALLOCATE PRIME   ON SP72 SIZE 440256K,
               OVERFLOW ON SP72 SIZE 1760K;

CREATE DSI ORDERLIN_73_DSI
     DSO ORDERLINE_DSO
       USING(793,803)
         ALLOCATE PRIME   ON SP73 SIZE 440256K,
               OVERFLOW ON SP73 SIZE 1760K;

CREATE DSI ORDERLIN_74_DSI
     DSO ORDERLINE_DSO
       USING(804,814)
         ALLOCATE PRIME   ON SP74 SIZE 440256K,
               OVERFLOW ON SP74 SIZE 1760K;

CREATE DSI ORDERLIN_75_DSI
     DSO ORDERLINE_DSO
       USING(815,825)
         ALLOCATE PRIME   ON SP75 SIZE 440256K,
               OVERFLOW ON SP75 SIZE 1760K;

CREATE DSI ORDERLIN_76_DSI
     DSO ORDERLINE_DSO
       USING(826,836)
         ALLOCATE PRIME   ON SP76 SIZE 440256K,
               OVERFLOW ON SP76 SIZE 1760K;

CREATE DSI ORDERLIN_77_DSI
     DSO ORDERLINE_DSO
       USING(837,847)
         ALLOCATE PRIME   ON SP77 SIZE 440256K,
               OVERFLOW ON SP77 SIZE 1760K;

CREATE DSI ORDERLIN_78_DSI
     DSO ORDERLINE_DSO
       USING(848,858)
         ALLOCATE PRIME   ON SP78 SIZE 440256K,
               OVERFLOW ON SP78 SIZE 1760K;

CREATE DSI ORDERLIN_79_DSI
     DSO ORDERLINE_DSO
       USING(859,869)
         ALLOCATE PRIME   ON SP79 SIZE 440256K,
               OVERFLOW ON SP79 SIZE 1760K;

CREATE DSI ORDERLIN_80_DSI

     DSO ORDERLINE_DSO
       USING(870,880)
         ALLOCATE PRIME   ON SP80 SIZE 440256K,
               OVERFLOW ON SP80 SIZE 1760K;

CREATE DSI ORDERLIN_81_DSI
     DSO ORDERLINE_DSO
       USING(881,891)
         ALLOCATE PRIME   ON SP81 SIZE 440256K,
               OVERFLOW ON SP81 SIZE 1760K;

CREATE DSI ORDERLIN_82_DSI
     DSO ORDERLINE_DSO
       USING(892,902)
         ALLOCATE PRIME   ON SP82 SIZE 440256K,
               OVERFLOW ON SP82 SIZE 1760K;

CREATE DSI ORDERLIN_83_DSI
     DSO ORDERLINE_DSO
       USING(903,913)
         ALLOCATE PRIME   ON SP83 SIZE 440256K,
               OVERFLOW ON SP83 SIZE 1760K;

CREATE DSI ORDERLIN_84_DSI
     DSO ORDERLINE_DSO
       USING(914,924)
         ALLOCATE PRIME   ON SP84 SIZE 440256K,
               OVERFLOW ON SP84 SIZE 1760K;

CREATE DSI ORDERLIN_85_DSI
     DSO ORDERLINE_DSO
       USING(925,935)
         ALLOCATE PRIME   ON SP85 SIZE 440256K,
               OVERFLOW ON SP85 SIZE 1760K;

CREATE DSI ORDERLIN_86_DSI
     DSO ORDERLINE_DSO
       USING(936,946)
         ALLOCATE PRIME   ON SP86 SIZE 440256K,
               OVERFLOW ON SP86 SIZE 1760K;

CREATE DSI ORDERLIN_87_DSI
     DSO ORDERLINE_DSO
       USING(947,957)
         ALLOCATE PRIME   ON SP87 SIZE 440256K,
               OVERFLOW ON SP87 SIZE 1760K;

CREATE DSI ORDERLIN_88_DSI
     DSO ORDERLINE_DSO
       USING(958,968)
         ALLOCATE PRIME   ON SP88 SIZE 440256K,
               OVERFLOW ON SP88 SIZE 1760K;

CREATE DSI ORDERLIN_89_DSI
     DSO ORDERLINE_DSO
       USING(969,979)
         ALLOCATE PRIME   ON SP89 SIZE 440256K,
               OVERFLOW ON SP89 SIZE 1760K;

CREATE DSI ORDERLIN_90_DSI
     DSO ORDERLINE_DSO
       USING(980,990)
         ALLOCATE PRIME   ON SP90 SIZE 440256K,

OVERFLOW ON SP90 SIZE 1760K;

CREATE DSI ORDERLIN_91_DSI
   DSO ORDERLINE_DSO
    USING(991,1001)
     ALLOCATE PRIME   ON SP91 SIZE 440256K,
       OVERFLOW ON SP91 SIZE 1760K;

CREATE DSI ORDERLIN_92_DSI
   DSO ORDERLINE_DSO
    USING(1002,1012)
     ALLOCATE PRIME   ON SP92 SIZE 440256K,
       OVERFLOW ON SP92 SIZE 1760K;

CREATE DSI ORDERLIN_93_DSI
   DSO ORDERLINE_DSO
    USING(1013,1023)
     ALLOCATE PRIME   ON SP93 SIZE 440256K,
       OVERFLOW ON SP93 SIZE 1760K;

CREATE DSI ORDERLIN_94_DSI
   DSO ORDERLINE_DSO
    USING(1024,1034)
     ALLOCATE PRIME   ON SP94 SIZE 440256K,
       OVERFLOW ON SP94 SIZE 1760K;

CREATE DSI ORDERLIN_95_DSI
   DSO ORDERLINE_DSO
    USING(1035,1045)
     ALLOCATE PRIME   ON SP95 SIZE 440256K,
       OVERFLOW ON SP95 SIZE 1760K;

CREATE DSI ORDERLIN_96_DSI
   DSO ORDERLINE_DSO
    USING(1046,1056)
     ALLOCATE PRIME   ON SP96 SIZE 440256K,
       OVERFLOW ON SP96 SIZE 1760K;

CREATE DSI ORDERLIN_97_DSI
   DSO ORDERLINE_DSO
    USING(1057,1067)
     ALLOCATE PRIME   ON SP97 SIZE 440256K,
       OVERFLOW ON SP97 SIZE 1760K;

CREATE DSI ORDERLIN_98_DSI
   DSO ORDERLINE_DSO
    USING(1068,1078)
     ALLOCATE PRIME   ON SP98 SIZE 440256K,
       OVERFLOW ON SP98 SIZE 1760K;

CREATE DSI ORDERLIN_99_DSI
   DSO ORDERLINE_DSO
    USING(1079,1089)
     ALLOCATE PRIME   ON SP99 SIZE 440256K,
       OVERFLOW ON SP99 SIZE 1760K;

CREATE DSI ORDERLIN_100_DSI
   DSO ORDERLINE_DSO
    USING(1090,1100)
     ALLOCATE PRIME   ON SP100 SIZE 440256K,
       OVERFLOW ON SP100 SIZE 1760K;

CREATE DSI ORDERLIN_101_DSI

DSO ORDERLINE_DSO
    USING(1101,1111)
     ALLOCATE PRIME   ON SP101 SIZE 440256K,
       OVERFLOW ON SP101 SIZE 1760K;

CREATE DSI ORDERLIN_102_DSI
   DSO ORDERLINE_DSO
    USING(1112,1122)
     ALLOCATE PRIME   ON SP102 SIZE 440256K,
       OVERFLOW ON SP102 SIZE 1760K;

CREATE DSI ORDERLIN_103_DSI
   DSO ORDERLINE_DSO
    USING(1123,1133)
     ALLOCATE PRIME   ON SP103 SIZE 440256K,
       OVERFLOW ON SP103 SIZE 1760K;

CREATE DSI ORDERLIN_104_DSI
   DSO ORDERLINE_DSO
    USING(1134,1144)
     ALLOCATE PRIME   ON SP104 SIZE 440256K,
       OVERFLOW ON SP104 SIZE 1760K;

CREATE DSI ORDERLIN_105_DSI
   DSO ORDERLINE_DSO
    USING(1145,1155)
     ALLOCATE PRIME   ON SP105 SIZE 440256K,
       OVERFLOW ON SP105 SIZE 1760K;

CREATE DSI ORDERLIN_106_DSI
   DSO ORDERLINE_DSO
    USING(1156,1166)
     ALLOCATE PRIME   ON SP106 SIZE 440256K,
       OVERFLOW ON SP106 SIZE 1760K;

CREATE DSI ORDERLIN_107_DSI
   DSO ORDERLINE_DSO
    USING(1167,1177)
     ALLOCATE PRIME   ON SP107 SIZE 440256K,
       OVERFLOW ON SP107 SIZE 1760K;

CREATE DSI ORDERLIN_108_DSI
   DSO ORDERLINE_DSO
    USING(1178,1188)
     ALLOCATE PRIME   ON SP108 SIZE 440256K,
       OVERFLOW ON SP108 SIZE 1760K;

CREATE DSI ORDERLIN_109_DSI
   DSO ORDERLINE_DSO
    USING(1189,1199)
     ALLOCATE PRIME   ON SP109 SIZE 440256K,
       OVERFLOW ON SP109 SIZE 1760K;

CREATE DSI ORDERLIN_110_DSI
   DSO ORDERLINE_DSO
    USING(1200,1210)
     ALLOCATE PRIME   ON SP110 SIZE 440256K,
       OVERFLOW ON SP110 SIZE 1760K;

CREATE DSI ORDERLIN_111_DSI
   DSO ORDERLINE_DSO
    USING(1211,1221)
     ALLOCATE PRIME   ON SP111 SIZE 440256K,

OVERFLOW ON SP111 SIZE 1760K;

CREATE DSI ORDERLIN_112_DSI
    DSO ORDERLINE_DSO
     USING(1222,1232)
     ALLOCATE PRIME   ON SP112 SIZE 440256K,
         OVERFLOW ON SP112 SIZE 1760K;

CREATE DSI ORDERLIN_113_DSI
    DSO ORDERLINE_DSO
     USING(1233,1243)
     ALLOCATE PRIME   ON SP113 SIZE 440256K,
         OVERFLOW ON SP113 SIZE 1760K;

CREATE DSI ORDERLIN_114_DSI
    DSO ORDERLINE_DSO
     USING(1244,1254)
     ALLOCATE PRIME   ON SP114 SIZE 440256K,
         OVERFLOW ON SP114 SIZE 1760K;

CREATE DSI ORDERLIN_115_DSI
    DSO ORDERLINE_DSO
     USING(1255,1265)
     ALLOCATE PRIME   ON SP115 SIZE 440256K,
         OVERFLOW ON SP115 SIZE 1760K;

CREATE DSI ORDERLIN_116_DSI
    DSO ORDERLINE_DSO
     USING(1266,1276)
     ALLOCATE PRIME   ON SP116 SIZE 440256K,
         OVERFLOW ON SP116 SIZE 1760K;

CREATE DSI ORDERLIN_117_DSI
    DSO ORDERLINE_DSO
     USING(1277,1287)
     ALLOCATE PRIME   ON SP117 SIZE 440256K,
         OVERFLOW ON SP117 SIZE 1760K;

CREATE DSI ORDERLIN_118_DSI
    DSO ORDERLINE_DSO
     USING(1288,1298)
     ALLOCATE PRIME   ON SP118 SIZE 440256K,
         OVERFLOW ON SP118 SIZE 1760K;

CREATE DSI ORDERLIN_119_DSI
    DSO ORDERLINE_DSO
     USING(1299,1309)
     ALLOCATE PRIME   ON SP119 SIZE 440256K,
         OVERFLOW ON SP119 SIZE 1760K;

CREATE DSI ORDERLIN_120_DSI
    DSO ORDERLINE_DSO
     USING(1310,1320)
     ALLOCATE PRIME   ON SP120 SIZE 440256K,
         OVERFLOW ON SP120 SIZE 1760K;

CREATE DSI ORDERLIN_121_DSI
    DSO ORDERLINE_DSO
     USING(1321,1331)
     ALLOCATE PRIME   ON SP121 SIZE 440256K,
         OVERFLOW ON SP121 SIZE 1760K;

CREATE DSI ORDERLIN_122_DSI

DSO ORDERLINE_DSO
     USING(1332,1342)
     ALLOCATE PRIME   ON SP122 SIZE 440256K,
         OVERFLOW ON SP122 SIZE 1760K;

CREATE DSI ORDERLIN_123_DSI
    DSO ORDERLINE_DSO
     USING(1343,1353)
     ALLOCATE PRIME   ON SP123 SIZE 440256K,
         OVERFLOW ON SP123 SIZE 1760K;

CREATE DSI ORDERLIN_124_DSI
    DSO ORDERLINE_DSO
     USING(1354,1364)
     ALLOCATE PRIME   ON SP124 SIZE 440256K,
         OVERFLOW ON SP124 SIZE 1760K;

CREATE DSI ORDERLIN_125_DSI
    DSO ORDERLINE_DSO
     USING(1365,1375)
     ALLOCATE PRIME   ON SP125 SIZE 440256K,
         OVERFLOW ON SP125 SIZE 1760K;

CREATE DSI ORDERLIN_126_DSI
    DSO ORDERLINE_DSO
     USING(1376,1386)
     ALLOCATE PRIME   ON SP126 SIZE 440256K,
         OVERFLOW ON SP126 SIZE 1760K;

CREATE DSI ORDERLIN_127_DSI
    DSO ORDERLINE_DSO
     USING(1387,1397)
     ALLOCATE PRIME   ON SP127 SIZE 440256K,
         OVERFLOW ON SP127 SIZE 1760K;

CREATE DSI ORDERLIN_128_DSI
    DSO ORDERLINE_DSO
     USING(1398,1408)
     ALLOCATE PRIME   ON SP128 SIZE 440256K,
         OVERFLOW ON SP128 SIZE 1760K;

CREATE DSI ORDERLIN_129_DSI
    DSO ORDERLINE_DSO
     USING(1409,1419)
     ALLOCATE PRIME   ON SP129 SIZE 440256K,
         OVERFLOW ON SP129 SIZE 1760K;

CREATE DSI ORDERLIN_130_DSI
    DSO ORDERLINE_DSO
     USING(1420,1430)
     ALLOCATE PRIME   ON SP130 SIZE 440256K,
         OVERFLOW ON SP130 SIZE 1760K;

CREATE DSI ORDERLIN_131_DSI
    DSO ORDERLINE_DSO
     USING(1431,1441)
     ALLOCATE PRIME   ON SP131 SIZE 440256K,
         OVERFLOW ON SP131 SIZE 1760K;

CREATE DSI ORDERLIN_132_DSI
    DSO ORDERLINE_DSO
     USING(1442,1452)
     ALLOCATE PRIME   ON SP132 SIZE 440256K,

OVERFLOW ON SP132 SIZE 1760K;

CREATE DSI ORDERLIN_133_DSI
    DSO ORDERLINE_DSO
     USING(1453,1463)
     ALLOCATE PRIME   ON SP133 SIZE 440256K,
        OVERFLOW ON SP133 SIZE 1760K;

CREATE DSI ORDERLIN_134_DSI
    DSO ORDERLINE_DSO
     USING(1464,1474)
     ALLOCATE PRIME   ON SP134 SIZE 440256K,
        OVERFLOW ON SP134 SIZE 1760K;

CREATE DSI ORDERLIN_135_DSI
    DSO ORDERLINE_DSO
     USING(1475,1485)
     ALLOCATE PRIME   ON SP135 SIZE 440256K,
        OVERFLOW ON SP135 SIZE 1760K;

CREATE DSI ORDERLIN_136_DSI
    DSO ORDERLINE_DSO
     USING(1486,1496)
     ALLOCATE PRIME   ON SP136 SIZE 440256K,
        OVERFLOW ON SP136 SIZE 1760K;

CREATE DSI ORDERLIN_137_DSI
    DSO ORDERLINE_DSO
     USING(1497,1507)
     ALLOCATE PRIME   ON SP137 SIZE 440256K,
        OVERFLOW ON SP137 SIZE 1760K;

CREATE DSI ORDERLIN_138_DSI
    DSO ORDERLINE_DSO
     USING(1508,1518)
     ALLOCATE PRIME   ON SP138 SIZE 440256K,
        OVERFLOW ON SP138 SIZE 1760K;

CREATE DSI ORDERLIN_139_DSI
    DSO ORDERLINE_DSO
     USING(1519,1529)
     ALLOCATE PRIME   ON SP139 SIZE 440256K,
        OVERFLOW ON SP139 SIZE 1760K;

CREATE DSI ORDERLIN_140_DSI
    DSO ORDERLINE_DSO
     USING(1530,1540)
     ALLOCATE PRIME   ON SP140 SIZE 440256K,
        OVERFLOW ON SP140 SIZE 1760K;

CREATE DSI ORDERLIN_141_DSI
    DSO ORDERLINE_DSO
     USING(1541,1551)
     ALLOCATE PRIME   ON SP141 SIZE 440256K,
        OVERFLOW ON SP141 SIZE 1760K;

CREATE DSI ORDERLIN_142_DSI
    DSO ORDERLINE_DSO
     USING(1552,1562)
     ALLOCATE PRIME   ON SP142 SIZE 440256K,
        OVERFLOW ON SP142 SIZE 1760K;

CREATE DSI ORDERLIN_143_DSI

DSO ORDERLINE_DSO
     USING(1563,1573)
     ALLOCATE PRIME   ON SP143 SIZE 440256K,
        OVERFLOW ON SP143 SIZE 1760K;

CREATE DSI ORDERLIN_144_DSI
    DSO ORDERLINE_DSO
     USING(1574,1584)
     ALLOCATE PRIME   ON SP144 SIZE 440256K,
        OVERFLOW ON SP144 SIZE 1760K;

CREATE DSI ORDERLIN_145_DSI
    DSO ORDERLINE_DSO
     USING(1585,1595)
     ALLOCATE PRIME   ON SP145 SIZE 440256K,
        OVERFLOW ON SP145 SIZE 1760K;

CREATE DSI ORDERLIN_146_DSI
    DSO ORDERLINE_DSO
     USING(1596,1606)
     ALLOCATE PRIME   ON SP146 SIZE 440256K,
        OVERFLOW ON SP146 SIZE 1760K;

CREATE DSI ORDERLIN_147_DSI
    DSO ORDERLINE_DSO
     USING(1607,1617)
     ALLOCATE PRIME   ON SP147 SIZE 440256K,
        OVERFLOW ON SP147 SIZE 1760K;

CREATE DSI ORDERLIN_148_DSI
    DSO ORDERLINE_DSO
     USING(1618,1628)
     ALLOCATE PRIME   ON SP148 SIZE 440256K,
        OVERFLOW ON SP148 SIZE 1760K;

CREATE DSI ORDERLIN_149_DSI
    DSO ORDERLINE_DSO
     USING(1629,1639)
     ALLOCATE PRIME   ON SP149 SIZE 440256K,
        OVERFLOW ON SP149 SIZE 1760K;

CREATE DSI ORDERLIN_150_DSI
    DSO ORDERLINE_DSO
     USING(1640,1650)
     ALLOCATE PRIME   ON SP150 SIZE 440256K,
        OVERFLOW ON SP150 SIZE 1760K;

CREATE DSI ORDERLIN_151_DSI
    DSO ORDERLINE_DSO
     USING(1651,1661)
     ALLOCATE PRIME   ON SP151 SIZE 440256K,
        OVERFLOW ON SP151 SIZE 1760K;

CREATE DSI ORDERLIN_152_DSI
    DSO ORDERLINE_DSO
     USING(1662,1672)
     ALLOCATE PRIME   ON SP152 SIZE 440256K,
        OVERFLOW ON SP152 SIZE 1760K;

CREATE DSI ORDERLIN_153_DSI
    DSO ORDERLINE_DSO
     USING(1673,1683)
     ALLOCATE PRIME   ON SP153 SIZE 440256K,

OVERFLOW ON SP153 SIZE 1760K;

CREATE DSI ORDERLIN_154_DSI
    DSO ORDERLINE_DSO
     USING(1684,1694)
     ALLOCATE PRIME   ON SP154 SIZE 440256K,
        OVERFLOW ON SP154 SIZE 1760K;

CREATE DSI ORDERLIN_155_DSI
    DSO ORDERLINE_DSO
     USING(1695,1705)
     ALLOCATE PRIME   ON SP155 SIZE 440256K,
        OVERFLOW ON SP155 SIZE 1760K;

CREATE DSI ORDERLIN_156_DSI
    DSO ORDERLINE_DSO
     USING(1706,1716)
     ALLOCATE PRIME   ON SP156 SIZE 440256K,
        OVERFLOW ON SP156 SIZE 1760K;

CREATE DSI ORDERLIN_157_DSI
    DSO ORDERLINE_DSO
     USING(1717,1727)
     ALLOCATE PRIME   ON SP157 SIZE 440256K,
        OVERFLOW ON SP157 SIZE 1760K;

CREATE DSI ORDERLIN_158_DSI
    DSO ORDERLINE_DSO
     USING(1728,1738)
     ALLOCATE PRIME   ON SP158 SIZE 440256K,
        OVERFLOW ON SP158 SIZE 1760K;

CREATE DSI ORDERLIN_159_DSI
    DSO ORDERLINE_DSO
     USING(1739,1749)
     ALLOCATE PRIME   ON SP159 SIZE 440256K,
        OVERFLOW ON SP159 SIZE 1760K;

CREATE DSI ORDERLIN_160_DSI
    DSO ORDERLINE_DSO
     USING(1750,1760)
     ALLOCATE PRIME   ON SP160 SIZE 440256K,
        OVERFLOW ON SP160 SIZE 1760K;

CREATE DSI ORDERLIN_161_DSI
    DSO ORDERLINE_DSO
     USING(1761,1771)
     ALLOCATE PRIME   ON SP161 SIZE 440256K,
        OVERFLOW ON SP161 SIZE 1760K;

CREATE DSI ORDERLIN_162_DSI
    DSO ORDERLINE_DSO
     USING(1772,1782)
     ALLOCATE PRIME   ON SP162 SIZE 440256K,
        OVERFLOW ON SP162 SIZE 1760K;

CREATE DSI ORDERLIN_163_DSI
    DSO ORDERLINE_DSO
     USING(1783,1793)
     ALLOCATE PRIME   ON SP163 SIZE 440256K,
        OVERFLOW ON SP163 SIZE 1760K;

CREATE DSI ORDERLIN_164_DSI

DSO ORDERLINE_DSO
     USING(1794,1804)
     ALLOCATE PRIME   ON SP164 SIZE 440256K,
        OVERFLOW ON SP164 SIZE 1760K;

CREATE DSI ORDERLIN_165_DSI
    DSO ORDERLINE_DSO
     USING(1805,1815)
     ALLOCATE PRIME   ON SP165 SIZE 440256K,
        OVERFLOW ON SP165 SIZE 1760K;

CREATE DSI ORDERLIN_166_DSI
    DSO ORDERLINE_DSO
     USING(1816,1826)
     ALLOCATE PRIME   ON SP166 SIZE 440256K,
        OVERFLOW ON SP166 SIZE 1760K;

CREATE DSI ORDERLIN_167_DSI
    DSO ORDERLINE_DSO
     USING(1827,1837)
     ALLOCATE PRIME   ON SP167 SIZE 440256K,
        OVERFLOW ON SP167 SIZE 1760K;

CREATE DSI ORDERLIN_168_DSI
    DSO ORDERLINE_DSO
     USING(1838,1848)
     ALLOCATE PRIME   ON SP168 SIZE 440256K,
        OVERFLOW ON SP168 SIZE 1760K;

CREATE DSI ORDERLIN_169_DSI
    DSO ORDERLINE_DSO
     USING(1849,1859)
     ALLOCATE PRIME   ON SP169 SIZE 440256K,
        OVERFLOW ON SP169 SIZE 1760K;

CREATE DSI ORDERLIN_170_DSI
    DSO ORDERLINE_DSO
     USING(1860,1870)
     ALLOCATE PRIME   ON SP170 SIZE 440256K,
        OVERFLOW ON SP170 SIZE 1760K;

CREATE DSI ORDERLIN_171_DSI
    DSO ORDERLINE_DSO
     USING(1871,1881)
     ALLOCATE PRIME   ON SP171 SIZE 440256K,
        OVERFLOW ON SP171 SIZE 1760K;

CREATE DSI ORDERLIN_172_DSI
    DSO ORDERLINE_DSO
     USING(1882,1892)
     ALLOCATE PRIME   ON SP172 SIZE 440256K,
        OVERFLOW ON SP172 SIZE 1760K;

CREATE DSI ORDERLIN_173_DSI
    DSO ORDERLINE_DSO
     USING(1893,1903)
     ALLOCATE PRIME   ON SP173 SIZE 440256K,
        OVERFLOW ON SP173 SIZE 1760K;

CREATE DSI ORDERLIN_174_DSI
    DSO ORDERLINE_DSO
     USING(1904,1914)
     ALLOCATE PRIME   ON SP174 SIZE 440256K,

```
                    OVERFLOW ON SP174 SIZE  1760K;              DSO ORDERLINE_DSO
                                                                USING(2025,2035)
CREATE DSI ORDERLIN_175_DSI                                      ALLOCATE PRIME    ON SP185 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP185 SIZE  1760K;
        USING(1915,1925)
        ALLOCATE PRIME    ON SP175 SIZE 440256K,       CREATE DSI ORDERLIN_186_DSI
              OVERFLOW ON SP175 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2036,2046)
CREATE DSI ORDERLIN_176_DSI                                      ALLOCATE PRIME    ON SP186 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP186 SIZE  1760K;
        USING(1926,1936)
        ALLOCATE PRIME    ON SP176 SIZE 440256K,       CREATE DSI ORDERLIN_187_DSI
              OVERFLOW ON SP176 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2047,2057)
CREATE DSI ORDERLIN_177_DSI                                      ALLOCATE PRIME    ON SP187 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP187 SIZE  1760K;
        USING(1937,1947)
        ALLOCATE PRIME    ON SP177 SIZE 440256K,       CREATE DSI ORDERLIN_188_DSI
              OVERFLOW ON SP177 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2058,2068)
CREATE DSI ORDERLIN_178_DSI                                      ALLOCATE PRIME    ON SP188 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP188 SIZE  1760K;
        USING(1948,1958)
        ALLOCATE PRIME    ON SP178 SIZE 440256K,       CREATE DSI ORDERLIN_189_DSI
              OVERFLOW ON SP178 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2069,2079)
CREATE DSI ORDERLIN_179_DSI                                      ALLOCATE PRIME    ON SP189 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP189 SIZE  1760K;
        USING(1959,1969)
        ALLOCATE PRIME    ON SP179 SIZE 440256K,       CREATE DSI ORDERLIN_190_DSI
              OVERFLOW ON SP179 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2080,2090)
CREATE DSI ORDERLIN_180_DSI                                      ALLOCATE PRIME    ON SP190 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP190 SIZE  1760K;
        USING(1970,1980)
        ALLOCATE PRIME    ON SP180 SIZE 440256K,       CREATE DSI ORDERLIN_191_DSI
              OVERFLOW ON SP180 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2091,2101)
CREATE DSI ORDERLIN_181_DSI                                      ALLOCATE PRIME    ON SP191 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP191 SIZE  1760K;
        USING(1981,1991)
        ALLOCATE PRIME    ON SP181 SIZE 440256K,       CREATE DSI ORDERLIN_192_DSI
              OVERFLOW ON SP181 SIZE  1760K;                 DSO ORDERLINE_DSO
                                                                USING(2102,4224)
CREATE DSI ORDERLIN_182_DSI                                      ALLOCATE PRIME    ON SP192 SIZE 440256K,
      DSO ORDERLINE_DSO                                               OVERFLOW ON SP192 SIZE  1760K;
        USING(1992,2002)
        ALLOCATE PRIME    ON SP182 SIZE 440256K,       ::::::::::::::
              OVERFLOW ON SP182 SIZE  1760K;           ddl.dat.OS
                                                       ::::::::::::::
CREATE DSI ORDERLIN_183_DSI                            --------------------------------------------------------------------------
      DSO ORDERLINE_DSO                                -- * Phase.2-4a: Orders
        USING(2003,2013)                               --------------------------------------------------------------------------
        ALLOCATE PRIME    ON SP183 SIZE 440256K,              CREATE DSO ORDERS_DSO
              OVERFLOW ON SP183 SIZE  1760K;                       FROM TPCC_SCHEMA.ORDERS
                                                                   TYPE
CREATE DSI ORDERLIN_184_DSI                            RANDOM(PAGESIZE1(8),PAGESIZE2(1),RULE((O_ID/12)*22+O_W_ID+((O_D_I
      DSO ORDERLINE_DSO                                D-1)*12+(O_ID-((O_ID/12)*12)))*6996))
        USING(2014,2024)                                         WHERE (O_W_ID) BETWEEN (?) AND (?);
        ALLOCATE PRIME    ON SP184 SIZE 440256K,
              OVERFLOW ON SP184 SIZE  1760K;                  CREATE DSI ORDERS_1_DSI
                                                                   DSO ORDERS_DSO
CREATE DSI ORDERLIN_185_DSI                                         USING(1,22)
```

```
        ALLOCATE PRIME    ON SP1 SIZE 27992K
                          SP2 SIZE 27984K,
        OVERFLOW ON SP1 SIZE 331K
                          SP2 SIZE 330K;

CREATE DSI ORDERS_2_DSI
    DSO ORDERS_DSO
    USING(23,44)
    ALLOCATE PRIME    ON SP3 SIZE 27992K
                      SP4 SIZE 27984K,
        OVERFLOW ON SP3 SIZE 331K
                      SP4 SIZE 330K;

CREATE DSI ORDERS_3_DSI
    DSO ORDERS_DSO
    USING(45,66)
    ALLOCATE PRIME    ON SP5 SIZE 27992K
                      SP6 SIZE 27984K,
        OVERFLOW ON SP5 SIZE 331K
                      SP6 SIZE 330K;

CREATE DSI ORDERS_4_DSI
    DSO ORDERS_DSO
    USING(67,88)
    ALLOCATE PRIME    ON SP7 SIZE 27992K
                      SP8 SIZE 27984K,
        OVERFLOW ON SP7 SIZE 331K
                      SP8 SIZE 330K;

CREATE DSI ORDERS_5_DSI
    DSO ORDERS_DSO
    USING(89,110)
    ALLOCATE PRIME    ON SP9 SIZE 27992K
                      SP10 SIZE 27984K,
        OVERFLOW ON SP9 SIZE 331K
                      SP10 SIZE 330K;

CREATE DSI ORDERS_6_DSI
    DSO ORDERS_DSO
    USING(111,132)
    ALLOCATE PRIME    ON SP11 SIZE 27992K
                      SP12 SIZE 27984K,
        OVERFLOW ON SP11 SIZE 331K
                      SP12 SIZE 330K;

CREATE DSI ORDERS_7_DSI
    DSO ORDERS_DSO
    USING(133,154)
    ALLOCATE PRIME    ON SP13 SIZE 27992K
                      SP14 SIZE 27984K,
        OVERFLOW ON SP13 SIZE 331K
                      SP14 SIZE 330K;

CREATE DSI ORDERS_8_DSI
    DSO ORDERS_DSO
    USING(155,176)
    ALLOCATE PRIME    ON SP15 SIZE 27992K
                      SP16 SIZE 27984K,
        OVERFLOW ON SP15 SIZE 331K
                      SP16 SIZE 330K;

CREATE DSI ORDERS_9_DSI
    DSO ORDERS_DSO
```

```
        USING(177,198)
        ALLOCATE PRIME    ON SP17 SIZE 27992K
                          SP18 SIZE 27984K,
            OVERFLOW ON SP17 SIZE 331K
                          SP18 SIZE 330K;

CREATE DSI ORDERS_10_DSI
    DSO ORDERS_DSO
    USING(199,220)
    ALLOCATE PRIME    ON SP19 SIZE 27992K
                      SP20 SIZE 27984K,
        OVERFLOW ON SP19 SIZE 331K
                      SP20 SIZE 330K;

CREATE DSI ORDERS_11_DSI
    DSO ORDERS_DSO
    USING(221,242)
    ALLOCATE PRIME    ON SP21 SIZE 27992K
                      SP22 SIZE 27984K,
        OVERFLOW ON SP21 SIZE 331K
                      SP22 SIZE 330K;

CREATE DSI ORDERS_12_DSI
    DSO ORDERS_DSO
    USING(243,264)
    ALLOCATE PRIME    ON SP23 SIZE 27992K
                      SP24 SIZE 27984K,
        OVERFLOW ON SP23 SIZE 331K
                      SP24 SIZE 330K;

CREATE DSI ORDERS_13_DSI
    DSO ORDERS_DSO
    USING(265,286)
    ALLOCATE PRIME    ON SP25 SIZE 27992K
                      SP26 SIZE 27984K,
        OVERFLOW ON SP25 SIZE 331K
                      SP26 SIZE 330K;

CREATE DSI ORDERS_14_DSI
    DSO ORDERS_DSO
    USING(287,308)
    ALLOCATE PRIME    ON SP27 SIZE 27992K
                      SP28 SIZE 27984K,
        OVERFLOW ON SP27 SIZE 331K
                      SP28 SIZE 330K;

CREATE DSI ORDERS_15_DSI
    DSO ORDERS_DSO
    USING(309,330)
    ALLOCATE PRIME    ON SP29 SIZE 27992K
                      SP30 SIZE 27984K,
        OVERFLOW ON SP29 SIZE 331K
                      SP30 SIZE 330K;

CREATE DSI ORDERS_16_DSI
    DSO ORDERS_DSO
    USING(331,352)
    ALLOCATE PRIME    ON SP31 SIZE 27992K
                      SP32 SIZE 27984K,
        OVERFLOW ON SP31 SIZE 331K
                      SP32 SIZE 330K;

CREATE DSI ORDERS_17_DSI
```

```
               DSO ORDERS_DSO
                USING(353,374)
                ALLOCATE PRIME    ON SP33 SIZE 27992K
                                      SP34 SIZE 27984K,
                        OVERFLOW ON SP33 SIZE 331K
                                      SP34 SIZE 330K;

CREATE DSI ORDERS_18_DSI
                DSO ORDERS_DSO
                USING(375,396)
                ALLOCATE PRIME    ON SP35 SIZE 27992K
                                      SP36 SIZE 27984K,
                        OVERFLOW ON SP35 SIZE 331K
                                      SP36 SIZE 330K;

CREATE DSI ORDERS_19_DSI
                DSO ORDERS_DSO
                USING(397,418)
                ALLOCATE PRIME    ON SP37 SIZE 27992K
                                      SP38 SIZE 27984K,
                        OVERFLOW ON SP37 SIZE 331K
                                      SP38 SIZE 330K;

CREATE DSI ORDERS_20_DSI
                DSO ORDERS_DSO
                USING(419,440)
                ALLOCATE PRIME    ON SP39 SIZE 27992K
                                      SP40 SIZE 27984K,
                        OVERFLOW ON SP39 SIZE 331K
                                      SP40 SIZE 330K;

CREATE DSI ORDERS_21_DSI
                DSO ORDERS_DSO
                USING(441,462)
                ALLOCATE PRIME    ON SP41 SIZE 27992K
                                      SP42 SIZE 27984K,
                        OVERFLOW ON SP41 SIZE 331K
                                      SP42 SIZE 330K;

CREATE DSI ORDERS_22_DSI
                DSO ORDERS_DSO
                USING(463,484)
                ALLOCATE PRIME    ON SP43 SIZE 27992K
                                      SP44 SIZE 27984K,
                        OVERFLOW ON SP43 SIZE 331K
                                      SP44 SIZE 330K;

CREATE DSI ORDERS_23_DSI
                DSO ORDERS_DSO
                USING(485,506)
                ALLOCATE PRIME    ON SP45 SIZE 27992K
                                      SP46 SIZE 27984K,
                        OVERFLOW ON SP45 SIZE 331K
                                      SP46 SIZE 330K;

CREATE DSI ORDERS_24_DSI
                DSO ORDERS_DSO
                USING(507,528)
                ALLOCATE PRIME    ON SP47 SIZE 27992K
                                      SP48 SIZE 27984K,
                        OVERFLOW ON SP47 SIZE 331K
                                      SP48 SIZE 330K;

CREATE DSI ORDERS_25_DSI
                DSO ORDERS_DSO
                USING(529,550)
                ALLOCATE PRIME    ON SP49 SIZE 27992K
                                      SP50 SIZE 27984K,
                        OVERFLOW ON SP49 SIZE 331K
                                      SP50 SIZE 330K;

CREATE DSI ORDERS_26_DSI
                DSO ORDERS_DSO
                USING(551,572)
                ALLOCATE PRIME    ON SP51 SIZE 27992K
                                      SP52 SIZE 27984K,
                        OVERFLOW ON SP51 SIZE 331K
                                      SP52 SIZE 330K;

CREATE DSI ORDERS_27_DSI
                DSO ORDERS_DSO
                USING(573,594)
                ALLOCATE PRIME    ON SP53 SIZE 27992K
                                      SP54 SIZE 27984K,
                        OVERFLOW ON SP53 SIZE 331K
                                      SP54 SIZE 330K;

CREATE DSI ORDERS_28_DSI
                DSO ORDERS_DSO
                USING(595,616)
                ALLOCATE PRIME    ON SP55 SIZE 27992K
                                      SP56 SIZE 27984K,
                        OVERFLOW ON SP55 SIZE 331K
                                      SP56 SIZE 330K;

CREATE DSI ORDERS_29_DSI
                DSO ORDERS_DSO
                USING(617,638)
                ALLOCATE PRIME    ON SP57 SIZE 27992K
                                      SP58 SIZE 27984K,
                        OVERFLOW ON SP57 SIZE 331K
                                      SP58 SIZE 330K;

CREATE DSI ORDERS_30_DSI
                DSO ORDERS_DSO
                USING(639,660)
                ALLOCATE PRIME    ON SP59 SIZE 27992K
                                      SP60 SIZE 27984K,
                        OVERFLOW ON SP59 SIZE 331K
                                      SP60 SIZE 330K;

CREATE DSI ORDERS_31_DSI
                DSO ORDERS_DSO
                USING(661,682)
                ALLOCATE PRIME    ON SP61 SIZE 27992K
                                      SP62 SIZE 27984K,
                        OVERFLOW ON SP61 SIZE 331K
                                      SP62 SIZE 330K;

CREATE DSI ORDERS_32_DSI
                DSO ORDERS_DSO
                USING(683,704)
                ALLOCATE PRIME    ON SP63 SIZE 27992K
                                      SP64 SIZE 27984K,
                        OVERFLOW ON SP63 SIZE 331K
                                      SP64 SIZE 330K;
```

```
                                                            SP80 SIZE 330K;
CREATE DSI ORDERS_33_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_41_DSI
   USING(705,726)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP65 SIZE 27992K      USING(881,902)
                      SP66 SIZE 27984K,          ALLOCATE PRIME    ON SP81 SIZE 27992K
        OVERFLOW ON SP65 SIZE 331K                                SP82 SIZE 27984K,
                      SP66 SIZE 330K;                  OVERFLOW ON SP81 SIZE 331K
                                                                  SP82 SIZE 330K;
CREATE DSI ORDERS_34_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_42_DSI
   USING(727,748)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP67 SIZE 27992K      USING(903,924)
                      SP68 SIZE 27984K,          ALLOCATE PRIME    ON SP83 SIZE 27992K
        OVERFLOW ON SP67 SIZE 331K                                SP84 SIZE 27984K,
                      SP68 SIZE 330K;                  OVERFLOW ON SP83 SIZE 331K
                                                                  SP84 SIZE 330K;
CREATE DSI ORDERS_35_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_43_DSI
   USING(749,770)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP69 SIZE 27992K      USING(925,946)
                      SP70 SIZE 27984K,          ALLOCATE PRIME    ON SP85 SIZE 27992K
        OVERFLOW ON SP69 SIZE 331K                                SP86 SIZE 27984K,
                      SP70 SIZE 330K;                  OVERFLOW ON SP85 SIZE 331K
                                                                  SP86 SIZE 330K;
CREATE DSI ORDERS_36_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_44_DSI
   USING(771,792)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP71 SIZE 27992K      USING(947,968)
                      SP72 SIZE 27984K,          ALLOCATE PRIME    ON SP87 SIZE 27992K
        OVERFLOW ON SP71 SIZE 331K                                SP88 SIZE 27984K,
                      SP72 SIZE 330K;                  OVERFLOW ON SP87 SIZE 331K
                                                                  SP88 SIZE 330K;
CREATE DSI ORDERS_37_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_45_DSI
   USING(793,814)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP73 SIZE 27992K      USING(969,990)
                      SP74 SIZE 27984K,          ALLOCATE PRIME    ON SP89 SIZE 27992K
        OVERFLOW ON SP73 SIZE 331K                                SP90 SIZE 27984K,
                      SP74 SIZE 330K;                  OVERFLOW ON SP89 SIZE 331K
                                                                  SP90 SIZE 330K;
CREATE DSI ORDERS_38_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_46_DSI
   USING(815,836)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP75 SIZE 27992K      USING(991,1012)
                      SP76 SIZE 27984K,          ALLOCATE PRIME    ON SP91 SIZE 27992K
        OVERFLOW ON SP75 SIZE 331K                                SP92 SIZE 27984K,
                      SP76 SIZE 330K;                  OVERFLOW ON SP91 SIZE 331K
                                                                  SP92 SIZE 330K;
CREATE DSI ORDERS_39_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_47_DSI
   USING(837,858)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP77 SIZE 27992K      USING(1013,1034)
                      SP78 SIZE 27984K,          ALLOCATE PRIME    ON SP93 SIZE 27992K
        OVERFLOW ON SP77 SIZE 331K                                SP94 SIZE 27984K,
                      SP78 SIZE 330K;                  OVERFLOW ON SP93 SIZE 331K
                                                                  SP94 SIZE 330K;
CREATE DSI ORDERS_40_DSI
   DSO ORDERS_DSO                        CREATE DSI ORDERS_48_DSI
   USING(859,880)                           DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP79 SIZE 27992K      USING(1035,1056)
                      SP80 SIZE 27984K,          ALLOCATE PRIME    ON SP95 SIZE 27992K
        OVERFLOW ON SP79 SIZE 331K                                SP96 SIZE 27984K,
```

```
                    OVERFLOW ON SP95 SIZE 331K                                                        SP112 SIZE 27984K,
                                    SP96 SIZE 330K;                         OVERFLOW ON SP111 SIZE 331K
                                                                                            SP112 SIZE 330K;
CREATE DSI ORDERS_49_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_57_DSI
    USING(1057,1078)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP97 SIZE 27992K                       USING(1233,1254)
                        SP98 SIZE 27984K,                       ALLOCATE PRIME    ON SP113 SIZE 27992K
        OVERFLOW ON SP97 SIZE 331K                                                  SP114 SIZE 27984K,
                        SP98 SIZE 330K;                             OVERFLOW ON SP113 SIZE 331K
                                                                                    SP114 SIZE 330K;
CREATE DSI ORDERS_50_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_58_DSI
    USING(1079,1100)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP99 SIZE 27992K                       USING(1255,1276)
                        SP100 SIZE 27984K,                      ALLOCATE PRIME    ON SP115 SIZE 27992K
        OVERFLOW ON SP99 SIZE 331K                                                  SP116 SIZE 27984K,
                        SP100 SIZE 330K;                          OVERFLOW ON SP115 SIZE 331K
                                                                                    SP116 SIZE 330K;
CREATE DSI ORDERS_51_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_59_DSI
    USING(1101,1122)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP101 SIZE 27992K                      USING(1277,1298)
                        SP102 SIZE 27984K,                      ALLOCATE PRIME    ON SP117 SIZE 27992K
        OVERFLOW ON SP101 SIZE 331K                                                 SP118 SIZE 27984K,
                        SP102 SIZE 330K;                          OVERFLOW ON SP117 SIZE 331K
                                                                                    SP118 SIZE 330K;
CREATE DSI ORDERS_52_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_60_DSI
    USING(1123,1144)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP103 SIZE 27992K                      USING(1299,1320)
                        SP104 SIZE 27984K,                      ALLOCATE PRIME    ON SP119 SIZE 27992K
        OVERFLOW ON SP103 SIZE 331K                                                 SP120 SIZE 27984K,
                        SP104 SIZE 330K;                          OVERFLOW ON SP119 SIZE 331K
                                                                                    SP120 SIZE 330K;
CREATE DSI ORDERS_53_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_61_DSI
    USING(1145,1166)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP105 SIZE 27992K                      USING(1321,1342)
                        SP106 SIZE 27984K,                      ALLOCATE PRIME    ON SP121 SIZE 27992K
        OVERFLOW ON SP105 SIZE 331K                                                 SP122 SIZE 27984K,
                        SP106 SIZE 330K;                          OVERFLOW ON SP121 SIZE 331K
                                                                                    SP122 SIZE 330K;
CREATE DSI ORDERS_54_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_62_DSI
    USING(1167,1188)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP107 SIZE 27992K                      USING(1343,1364)
                        SP108 SIZE 27984K,                      ALLOCATE PRIME    ON SP123 SIZE 27992K
        OVERFLOW ON SP107 SIZE 331K                                                 SP124 SIZE 27984K,
                        SP108 SIZE 330K;                          OVERFLOW ON SP123 SIZE 331K
                                                                                    SP124 SIZE 330K;
CREATE DSI ORDERS_55_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_63_DSI
    USING(1189,1210)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP109 SIZE 27992K                      USING(1365,1386)
                        SP110 SIZE 27984K,                      ALLOCATE PRIME    ON SP125 SIZE 27992K
        OVERFLOW ON SP109 SIZE 331K                                                 SP126 SIZE 27984K,
                        SP110 SIZE 330K;                          OVERFLOW ON SP125 SIZE 331K
                                                                                    SP126 SIZE 330K;
CREATE DSI ORDERS_56_DSI
    DSO ORDERS_DSO                                          CREATE DSI ORDERS_64_DSI
    USING(1211,1232)                                            DSO ORDERS_DSO
    ALLOCATE PRIME    ON SP111 SIZE 27992K                      USING(1387,1408)
```

```
            ALLOCATE PRIME    ON SP127 SIZE 27992K
                                     SP128 SIZE 27984K,
                  OVERFLOW ON SP127 SIZE 331K
                                     SP128 SIZE 330K;

CREATE DSI ORDERS_65_DSI
       DSO ORDERS_DSO
       USING(1409,1430)
         ALLOCATE PRIME    ON SP129 SIZE 27992K
                                     SP130 SIZE 27984K,
                  OVERFLOW ON SP129 SIZE 331K
                                     SP130 SIZE 330K;

CREATE DSI ORDERS_66_DSI
       DSO ORDERS_DSO
       USING(1431,1452)
         ALLOCATE PRIME    ON SP131 SIZE 27992K
                                     SP132 SIZE 27984K,
                  OVERFLOW ON SP131 SIZE 331K
                                     SP132 SIZE 330K;

CREATE DSI ORDERS_67_DSI
       DSO ORDERS_DSO
       USING(1453,1474)
         ALLOCATE PRIME    ON SP133 SIZE 27992K
                                     SP134 SIZE 27984K,
                  OVERFLOW ON SP133 SIZE 331K
                                     SP134 SIZE 330K;

CREATE DSI ORDERS_68_DSI
       DSO ORDERS_DSO
       USING(1475,1496)
         ALLOCATE PRIME    ON SP135 SIZE 27992K
                                     SP136 SIZE 27984K,
                  OVERFLOW ON SP135 SIZE 331K
                                     SP136 SIZE 330K;

CREATE DSI ORDERS_69_DSI
       DSO ORDERS_DSO
       USING(1497,1518)
         ALLOCATE PRIME    ON SP137 SIZE 27992K
                                     SP138 SIZE 27984K,
                  OVERFLOW ON SP137 SIZE 331K
                                     SP138 SIZE 330K;

CREATE DSI ORDERS_70_DSI
       DSO ORDERS_DSO
       USING(1519,1540)
         ALLOCATE PRIME    ON SP139 SIZE 27992K
                                     SP140 SIZE 27984K,
                  OVERFLOW ON SP139 SIZE 331K
                                     SP140 SIZE 330K;

CREATE DSI ORDERS_71_DSI
       DSO ORDERS_DSO
       USING(1541,1562)
         ALLOCATE PRIME    ON SP141 SIZE 27992K
                                     SP142 SIZE 27984K,
                  OVERFLOW ON SP141 SIZE 331K
                                     SP142 SIZE 330K;

CREATE DSI ORDERS_72_DSI
       DSO ORDERS_DSO


            USING(1563,1584)
         ALLOCATE PRIME    ON SP143 SIZE 27992K
                                     SP144 SIZE 27984K,
                  OVERFLOW ON SP143 SIZE 331K
                                     SP144 SIZE 330K;

CREATE DSI ORDERS_73_DSI
       DSO ORDERS_DSO
       USING(1585,1606)
         ALLOCATE PRIME    ON SP145 SIZE 27992K
                                     SP146 SIZE 27984K,
                  OVERFLOW ON SP145 SIZE 331K
                                     SP146 SIZE 330K;

CREATE DSI ORDERS_74_DSI
       DSO ORDERS_DSO
       USING(1607,1628)
         ALLOCATE PRIME    ON SP147 SIZE 27992K
                                     SP148 SIZE 27984K,
                  OVERFLOW ON SP147 SIZE 331K
                                     SP148 SIZE 330K;

CREATE DSI ORDERS_75_DSI
       DSO ORDERS_DSO
       USING(1629,1650)
         ALLOCATE PRIME    ON SP149 SIZE 27992K
                                     SP150 SIZE 27984K,
                  OVERFLOW ON SP149 SIZE 331K
                                     SP150 SIZE 330K;

CREATE DSI ORDERS_76_DSI
       DSO ORDERS_DSO
       USING(1651,1672)
         ALLOCATE PRIME    ON SP151 SIZE 27992K
                                     SP152 SIZE 27984K,
                  OVERFLOW ON SP151 SIZE 331K
                                     SP152 SIZE 330K;

CREATE DSI ORDERS_77_DSI
       DSO ORDERS_DSO
       USING(1673,1694)
         ALLOCATE PRIME    ON SP153 SIZE 27992K
                                     SP154 SIZE 27984K,
                  OVERFLOW ON SP153 SIZE 331K
                                     SP154 SIZE 330K;

CREATE DSI ORDERS_78_DSI
       DSO ORDERS_DSO
       USING(1695,1716)
         ALLOCATE PRIME    ON SP155 SIZE 27992K
                                     SP156 SIZE 27984K,
                  OVERFLOW ON SP155 SIZE 331K
                                     SP156 SIZE 330K;

CREATE DSI ORDERS_79_DSI
       DSO ORDERS_DSO
       USING(1717,1738)
         ALLOCATE PRIME    ON SP157 SIZE 27992K
                                     SP158 SIZE 27984K,
                  OVERFLOW ON SP157 SIZE 331K
                                     SP158 SIZE 330K;

CREATE DSI ORDERS_80_DSI
```

```
      DSO ORDERS_DSO
        USING(1739,1760)
        ALLOCATE PRIME    ON SP159 SIZE 27992K
                          SP160 SIZE 27984K,
            OVERFLOW ON SP159 SIZE 331K
                          SP160 SIZE 330K;

CREATE DSI ORDERS_81_DSI
      DSO ORDERS_DSO
        USING(1761,1782)
        ALLOCATE PRIME    ON SP161 SIZE 27992K
                          SP162 SIZE 27984K,
            OVERFLOW ON SP161 SIZE 331K
                          SP162 SIZE 330K;

CREATE DSI ORDERS_82_DSI
      DSO ORDERS_DSO
        USING(1783,1804)
        ALLOCATE PRIME    ON SP163 SIZE 27992K
                          SP164 SIZE 27984K,
            OVERFLOW ON SP163 SIZE 331K
                          SP164 SIZE 330K;

CREATE DSI ORDERS_83_DSI
      DSO ORDERS_DSO
        USING(1805,1826)
        ALLOCATE PRIME    ON SP165 SIZE 27992K
                          SP166 SIZE 27984K,
            OVERFLOW ON SP165 SIZE 331K
                          SP166 SIZE 330K;

CREATE DSI ORDERS_84_DSI
      DSO ORDERS_DSO
        USING(1827,1848)
        ALLOCATE PRIME    ON SP167 SIZE 27992K
                          SP168 SIZE 27984K,
            OVERFLOW ON SP167 SIZE 331K
                          SP168 SIZE 330K;

CREATE DSI ORDERS_85_DSI
      DSO ORDERS_DSO
        USING(1849,1870)
        ALLOCATE PRIME    ON SP169 SIZE 27992K
                          SP170 SIZE 27984K,
            OVERFLOW ON SP169 SIZE 331K
                          SP170 SIZE 330K;

CREATE DSI ORDERS_86_DSI
      DSO ORDERS_DSO
        USING(1871,1892)
        ALLOCATE PRIME    ON SP171 SIZE 27992K
                          SP172 SIZE 27984K,
            OVERFLOW ON SP171 SIZE 331K
                          SP172 SIZE 330K;

CREATE DSI ORDERS_87_DSI
      DSO ORDERS_DSO
        USING(1893,1914)
        ALLOCATE PRIME    ON SP173 SIZE 27992K
                          SP174 SIZE 27984K,
            OVERFLOW ON SP173 SIZE 331K
                          SP174 SIZE 330K;
```

```
CREATE DSI ORDERS_88_DSI
      DSO ORDERS_DSO
        USING(1915,1936)
        ALLOCATE PRIME    ON SP175 SIZE 27992K
                          SP176 SIZE 27984K,
            OVERFLOW ON SP175 SIZE 331K
                          SP176 SIZE 330K;

CREATE DSI ORDERS_89_DSI
      DSO ORDERS_DSO
        USING(1937,1958)
        ALLOCATE PRIME    ON SP177 SIZE 27992K
                          SP178 SIZE 27984K,
            OVERFLOW ON SP177 SIZE 331K
                          SP178 SIZE 330K;

CREATE DSI ORDERS_90_DSI
      DSO ORDERS_DSO
        USING(1959,1980)
        ALLOCATE PRIME    ON SP179 SIZE 27992K
                          SP180 SIZE 27984K,
            OVERFLOW ON SP179 SIZE 331K
                          SP180 SIZE 330K;

CREATE DSI ORDERS_91_DSI
      DSO ORDERS_DSO
        USING(1981,2002)
        ALLOCATE PRIME    ON SP181 SIZE 27992K
                          SP182 SIZE 27984K,
            OVERFLOW ON SP181 SIZE 331K
                          SP182 SIZE 330K;

CREATE DSI ORDERS_92_DSI
      DSO ORDERS_DSO
        USING(2003,2024)
        ALLOCATE PRIME    ON SP183 SIZE 27992K
                          SP184 SIZE 27984K,
            OVERFLOW ON SP183 SIZE 331K
                          SP184 SIZE 330K;

CREATE DSI ORDERS_93_DSI
      DSO ORDERS_DSO
        USING(2025,2046)
        ALLOCATE PRIME    ON SP185 SIZE 27992K
                          SP186 SIZE 27984K,
            OVERFLOW ON SP185 SIZE 331K
                          SP186 SIZE 330K;

CREATE DSI ORDERS_94_DSI
      DSO ORDERS_DSO
        USING(2047,2068)
        ALLOCATE PRIME    ON SP187 SIZE 27992K
                          SP188 SIZE 27984K,
            OVERFLOW ON SP187 SIZE 331K
                          SP188 SIZE 330K;

CREATE DSI ORDERS_95_DSI
      DSO ORDERS_DSO
        USING(2069,2090)
        ALLOCATE PRIME    ON SP189 SIZE 27992K
                          SP190 SIZE 27984K,
            OVERFLOW ON SP189 SIZE 331K
                          SP190 SIZE 330K;
```

```
        CREATE DSI ORDERS_96_DSI
            DSO ORDERS_DSO
            USING(2091,4224)
                ALLOCATE PRIME    ON SP191 SIZE 27992K
                                     SP192 SIZE 27984K,
                    OVERFLOW ON SP191 SIZE 331K
                                     SP192 SIZE 330K;

::::::::::::::
ddl.dat.ST
::::::::::::::
-----------------------------------------------------------------------------
-- * Phase.2-8: Stock
-----------------------------------------------------------------------------
        CREATE DSO STOCK_DSO
            FROM TPCC_SCHEMA.STOCK
                TYPE RANDOM(PAGESIZE1(4),PAGESIZE2(1),
                RULE(S_I_ID*4+(S_W_ID-1)/11+(S_W_ID-
S_W_ID/11*11)*400000))
                WHERE (S_W_ID) BETWEEN (?) AND (?);
        CREATE DSI STOCK_1_DSI
            DSO STOCK_DSO
            USING(1,44)
                ALLOCATE PRIME    ON SP1 SIZE 200004K
                                     SP2 SIZE 200000K
                                     SP3 SIZE 200000K
                                     SP4 SIZE 200000K
                                     SP5 SIZE 200000K
                                     SP6 SIZE 200000K
                                     SP7 SIZE 200000K
                                     SP8 SIZE 200000K,
                    OVERFLOW ON SP5 SIZE 80001K;

        CREATE DSI STOCK_2_DSI
            DSO STOCK_DSO
            USING(45,88)
                ALLOCATE PRIME    ON SP1 SIZE 200004K
                                     SP2 SIZE 200000K
                                     SP3 SIZE 200000K
                                     SP4 SIZE 200000K
                                     SP5 SIZE 200000K
                                     SP6 SIZE 200000K
                                     SP7 SIZE 200000K
                                     SP8 SIZE 200000K,
                    OVERFLOW ON SP5 SIZE 80001K;

        CREATE DSI STOCK_3_DSI
            DSO STOCK_DSO
            USING(89,132)
                ALLOCATE PRIME    ON SP9 SIZE 200004K
                                     SP10 SIZE 200000K
                                     SP11 SIZE 200000K
                                     SP12 SIZE 200000K
                                     SP13 SIZE 200000K
                                     SP14 SIZE 200000K
                                     SP15 SIZE 200000K
                                     SP16 SIZE 200000K,
                    OVERFLOW ON SP13 SIZE 80001K;

        CREATE DSI STOCK_4_DSI
            DSO STOCK_DSO
```

```
            USING(133,176)
                ALLOCATE PRIME    ON SP9 SIZE 200004K
                                     SP10 SIZE 200000K
                                     SP11 SIZE 200000K
                                     SP12 SIZE 200000K
                                     SP13 SIZE 200000K
                                     SP14 SIZE 200000K
                                     SP15 SIZE 200000K
                                     SP16 SIZE 200000K,
                    OVERFLOW ON SP13 SIZE 80001K;

        CREATE DSI STOCK_5_DSI
            DSO STOCK_DSO
            USING(177,220)
                ALLOCATE PRIME    ON SP17 SIZE 200004K
                                     SP18 SIZE 200000K
                                     SP19 SIZE 200000K
                                     SP20 SIZE 200000K
                                     SP21 SIZE 200000K
                                     SP22 SIZE 200000K
                                     SP23 SIZE 200000K
                                     SP24 SIZE 200000K,
                    OVERFLOW ON SP21 SIZE 80001K;

        CREATE DSI STOCK_6_DSI
            DSO STOCK_DSO
            USING(221,264)
                ALLOCATE PRIME    ON SP17 SIZE 200004K
                                     SP18 SIZE 200000K
                                     SP19 SIZE 200000K
                                     SP20 SIZE 200000K
                                     SP21 SIZE 200000K
                                     SP22 SIZE 200000K
                                     SP23 SIZE 200000K
                                     SP24 SIZE 200000K,
                    OVERFLOW ON SP21 SIZE 80001K;

        CREATE DSI STOCK_7_DSI
            DSO STOCK_DSO
            USING(265,308)
                ALLOCATE PRIME    ON SP25 SIZE 200004K
                                     SP26 SIZE 200000K
                                     SP27 SIZE 200000K
                                     SP28 SIZE 200000K
                                     SP29 SIZE 200000K
                                     SP30 SIZE 200000K
                                     SP31 SIZE 200000K
                                     SP32 SIZE 200000K,
                    OVERFLOW ON SP29 SIZE 80001K;

        CREATE DSI STOCK_8_DSI
            DSO STOCK_DSO
            USING(309,352)
                ALLOCATE PRIME    ON SP25 SIZE 200004K
                                     SP26 SIZE 200000K
                                     SP27 SIZE 200000K
                                     SP28 SIZE 200000K
                                     SP29 SIZE 200000K
                                     SP30 SIZE 200000K
                                     SP31 SIZE 200000K
                                     SP32 SIZE 200000K,
                    OVERFLOW ON SP29 SIZE 80001K;
```

```
CREATE DSI STOCK_9_DSI
     DSO STOCK_DSO
     USING(353,396)
     ALLOCATE PRIME    ON SP33 SIZE 200004K
                          SP34 SIZE 200000K
                          SP35 SIZE 200000K
                          SP36 SIZE 200000K
                          SP37 SIZE 200000K
                          SP38 SIZE 200000K
                          SP39 SIZE 200000K
                          SP40 SIZE 200000K,
            OVERFLOW ON SP37 SIZE 80001K;

CREATE DSI STOCK_10_DSI
     DSO STOCK_DSO
     USING(397,440)
     ALLOCATE PRIME    ON SP33 SIZE 200004K
                          SP34 SIZE 200000K
                          SP35 SIZE 200000K
                          SP36 SIZE 200000K
                          SP37 SIZE 200000K
                          SP38 SIZE 200000K
                          SP39 SIZE 200000K
                          SP40 SIZE 200000K,
            OVERFLOW ON SP37 SIZE 80001K;

CREATE DSI STOCK_11_DSI
     DSO STOCK_DSO
     USING(441,484)
     ALLOCATE PRIME    ON SP41 SIZE 200004K
                          SP42 SIZE 200000K
                          SP43 SIZE 200000K
                          SP44 SIZE 200000K
                          SP45 SIZE 200000K
                          SP46 SIZE 200000K
                          SP47 SIZE 200000K
                          SP48 SIZE 200000K,
            OVERFLOW ON SP45 SIZE 80001K;

CREATE DSI STOCK_12_DSI
     DSO STOCK_DSO
     USING(485,528)
     ALLOCATE PRIME    ON SP41 SIZE 200004K
                          SP42 SIZE 200000K
                          SP43 SIZE 200000K
                          SP44 SIZE 200000K
                          SP45 SIZE 200000K
                          SP46 SIZE 200000K
                          SP47 SIZE 200000K
                          SP48 SIZE 200000K,
            OVERFLOW ON SP45 SIZE 80001K;

CREATE DSI STOCK_13_DSI
     DSO STOCK_DSO
     USING(529,572)
     ALLOCATE PRIME    ON SP49 SIZE 200004K
                          SP50 SIZE 200000K
                          SP51 SIZE 200000K
                          SP52 SIZE 200000K
                          SP53 SIZE 200000K
                          SP54 SIZE 200000K
                          SP55 SIZE 200000K
                          SP56 SIZE 200000K,

            OVERFLOW ON SP53 SIZE 80001K;

CREATE DSI STOCK_14_DSI
     DSO STOCK_DSO
     USING(573,616)
     ALLOCATE PRIME    ON SP49 SIZE 200004K
                          SP50 SIZE 200000K
                          SP51 SIZE 200000K
                          SP52 SIZE 200000K
                          SP53 SIZE 200000K
                          SP54 SIZE 200000K
                          SP55 SIZE 200000K
                          SP56 SIZE 200000K,
            OVERFLOW ON SP53 SIZE 80001K;

CREATE DSI STOCK_15_DSI
     DSO STOCK_DSO
     USING(617,660)
     ALLOCATE PRIME    ON SP57 SIZE 200004K
                          SP58 SIZE 200000K
                          SP59 SIZE 200000K
                          SP60 SIZE 200000K
                          SP61 SIZE 200000K
                          SP62 SIZE 200000K
                          SP63 SIZE 200000K
                          SP64 SIZE 200000K,
            OVERFLOW ON SP61 SIZE 80001K;

CREATE DSI STOCK_16_DSI
     DSO STOCK_DSO
     USING(661,704)
     ALLOCATE PRIME    ON SP57 SIZE 200004K
                          SP58 SIZE 200000K
                          SP59 SIZE 200000K
                          SP60 SIZE 200000K
                          SP61 SIZE 200000K
                          SP62 SIZE 200000K
                          SP63 SIZE 200000K
                          SP64 SIZE 200000K,
            OVERFLOW ON SP61 SIZE 80001K;

CREATE DSI STOCK_17_DSI
     DSO STOCK_DSO
     USING(705,748)
     ALLOCATE PRIME    ON SP65 SIZE 200004K
                          SP66 SIZE 200000K
                          SP67 SIZE 200000K
                          SP68 SIZE 200000K
                          SP69 SIZE 200000K
                          SP70 SIZE 200000K
                          SP71 SIZE 200000K
                          SP72 SIZE 200000K,
            OVERFLOW ON SP69 SIZE 80001K;

CREATE DSI STOCK_18_DSI
     DSO STOCK_DSO
     USING(749,792)
     ALLOCATE PRIME    ON SP65 SIZE 200004K
                          SP66 SIZE 200000K
                          SP67 SIZE 200000K
                          SP68 SIZE 200000K
                          SP69 SIZE 200000K
                          SP70 SIZE 200000K
```

```
                              SP71 SIZE 200000K                                              SP93 SIZE 200000K
                              SP72 SIZE 200000K,                                             SP94 SIZE 200000K
             OVERFLOW ON SP69 SIZE 80001K;                                                   SP95 SIZE 200000K
                                                                                             SP96 SIZE 200000K,
CREATE DSI STOCK_19_DSI                                                      OVERFLOW ON SP93 SIZE 80001K;
      DSO STOCK_DSO
      USING(793,836)                                            CREATE DSI STOCK_24_DSI
      ALLOCATE PRIME    ON SP73 SIZE 200004K                          DSO STOCK_DSO
                              SP74 SIZE 200000K                       USING(1013,1056)
                              SP75 SIZE 200000K                       ALLOCATE PRIME    ON SP89 SIZE 200004K
                              SP76 SIZE 200000K                                               SP90 SIZE 200000K
                              SP77 SIZE 200000K                                               SP91 SIZE 200000K
                              SP78 SIZE 200000K                                               SP92 SIZE 200000K
                              SP79 SIZE 200000K                                               SP93 SIZE 200000K
                              SP80 SIZE 200000K,                                              SP94 SIZE 200000K
             OVERFLOW ON SP77 SIZE 80001K;                                                    SP95 SIZE 200000K
                                                                                             SP96 SIZE 200000K,
CREATE DSI STOCK_20_DSI                                                      OVERFLOW ON SP93 SIZE 80001K;
      DSO STOCK_DSO
      USING(837,880)                                           CREATE DSI STOCK_25_DSI
      ALLOCATE PRIME    ON SP73 SIZE 200004K                          DSO STOCK_DSO
                              SP74 SIZE 200000K                       USING(1057,1100)
                              SP75 SIZE 200000K                       ALLOCATE PRIME    ON SP97 SIZE 200004K
                              SP76 SIZE 200000K                                               SP98 SIZE 200000K
                              SP77 SIZE 200000K                                               SP99 SIZE 200000K
                              SP78 SIZE 200000K                                               SP100 SIZE 200000K
                              SP79 SIZE 200000K                                               SP101 SIZE 200000K
                              SP80 SIZE 200000K,                                              SP102 SIZE 200000K
             OVERFLOW ON SP77 SIZE 80001K;                                                    SP103 SIZE 200000K
                                                                                             SP104 SIZE 200000K,
CREATE DSI STOCK_21_DSI                                                      OVERFLOW ON SP101 SIZE 80001K;
      DSO STOCK_DSO
      USING(881,924)                                           CREATE DSI STOCK_26_DSI
      ALLOCATE PRIME    ON SP81 SIZE 200004K                          DSO STOCK_DSO
                              SP82 SIZE 200000K                       USING(1101,1144)
                              SP83 SIZE 200000K                       ALLOCATE PRIME    ON SP97 SIZE 200004K
                              SP84 SIZE 200000K                                               SP98 SIZE 200000K
                              SP85 SIZE 200000K                                               SP99 SIZE 200000K
                              SP86 SIZE 200000K                                               SP100 SIZE 200000K
                              SP87 SIZE 200000K                                               SP101 SIZE 200000K
                              SP88 SIZE 200000K,                                              SP102 SIZE 200000K
             OVERFLOW ON SP85 SIZE 80001K;                                                    SP103 SIZE 200000K
                                                                                             SP104 SIZE 200000K,
CREATE DSI STOCK_22_DSI                                                      OVERFLOW ON SP101 SIZE 80001K;
      DSO STOCK_DSO
      USING(925,968)                                           CREATE DSI STOCK_27_DSI
      ALLOCATE PRIME    ON SP81 SIZE 200004K                          DSO STOCK_DSO
                              SP82 SIZE 200000K                       USING(1145,1188)
                              SP83 SIZE 200000K                       ALLOCATE PRIME    ON SP105 SIZE 200004K
                              SP84 SIZE 200000K                                               SP106 SIZE 200000K
                              SP85 SIZE 200000K                                               SP107 SIZE 200000K
                              SP86 SIZE 200000K                                               SP108 SIZE 200000K
                              SP87 SIZE 200000K                                               SP109 SIZE 200000K
                              SP88 SIZE 200000K,                                              SP110 SIZE 200000K
             OVERFLOW ON SP85 SIZE 80001K;                                                    SP111 SIZE 200000K
                                                                                             SP112 SIZE 200000K,
CREATE DSI STOCK_23_DSI                                                      OVERFLOW ON SP109 SIZE 80001K;
      DSO STOCK_DSO
      USING(969,1012)                                          CREATE DSI STOCK_28_DSI
      ALLOCATE PRIME    ON SP89 SIZE 200004K                          DSO STOCK_DSO
                              SP90 SIZE 200000K                       USING(1189,1232)
                              SP91 SIZE 200000K                       ALLOCATE PRIME    ON SP105 SIZE 200004K
                              SP92 SIZE 200000K                                               SP106 SIZE 200000K
```

```
                         SP107 SIZE 200000K                    ALLOCATE PRIME    ON SP129 SIZE 200004K
                         SP108 SIZE 200000K                                      SP130 SIZE 200000K
                         SP109 SIZE 200000K                                      SP131 SIZE 200000K
                         SP110 SIZE 200000K                                      SP132 SIZE 200000K
                         SP111 SIZE 200000K                                      SP133 SIZE 200000K
                         SP112 SIZE 200000K,                                     SP134 SIZE 200000K
           OVERFLOW ON SP109 SIZE 80001K;                                        SP135 SIZE 200000K
                                                                                 SP136 SIZE 200000K,
CREATE DSI STOCK_29_DSI                                            OVERFLOW ON SP133 SIZE 80001K;
     DSO STOCK_DSO
     USING(1233,1276)                                       CREATE DSI STOCK_34_DSI
     ALLOCATE PRIME    ON SP113 SIZE 200004K                     DSO STOCK_DSO
                         SP114 SIZE 200000K                      USING(1453,1496)
                         SP115 SIZE 200000K                      ALLOCATE PRIME    ON SP129 SIZE 200004K
                         SP116 SIZE 200000K                                        SP130 SIZE 200000K
                         SP117 SIZE 200000K                                        SP131 SIZE 200000K
                         SP118 SIZE 200000K                                        SP132 SIZE 200000K
                         SP119 SIZE 200000K                                        SP133 SIZE 200000K
                         SP120 SIZE 200000K,                                       SP134 SIZE 200000K
           OVERFLOW ON SP117 SIZE 80001K;                                          SP135 SIZE 200000K
                                                                                   SP136 SIZE 200000K,
CREATE DSI STOCK_30_DSI                                              OVERFLOW ON SP133 SIZE 80001K;
     DSO STOCK_DSO
     USING(1277,1320)                                       CREATE DSI STOCK_35_DSI
     ALLOCATE PRIME    ON SP113 SIZE 200004K                     DSO STOCK_DSO
                         SP114 SIZE 200000K                      USING(1497,1540)
                         SP115 SIZE 200000K                      ALLOCATE PRIME    ON SP137 SIZE 200004K
                         SP116 SIZE 200000K                                        SP138 SIZE 200000K
                         SP117 SIZE 200000K                                        SP139 SIZE 200000K
                         SP118 SIZE 200000K                                        SP140 SIZE 200000K
                         SP119 SIZE 200000K                                        SP141 SIZE 200000K
                         SP120 SIZE 200000K,                                       SP142 SIZE 200000K
           OVERFLOW ON SP117 SIZE 80001K;                                          SP143 SIZE 200000K
                                                                                   SP144 SIZE 200000K,
CREATE DSI STOCK_31_DSI                                              OVERFLOW ON SP141 SIZE 80001K;
     DSO STOCK_DSO
     USING(1321,1364)                                       CREATE DSI STOCK_36_DSI
     ALLOCATE PRIME    ON SP121 SIZE 200004K                     DSO STOCK_DSO
                         SP122 SIZE 200000K                      USING(1541,1584)
                         SP123 SIZE 200000K                      ALLOCATE PRIME    ON SP137 SIZE 200004K
                         SP124 SIZE 200000K                                        SP138 SIZE 200000K
                         SP125 SIZE 200000K                                        SP139 SIZE 200000K
                         SP126 SIZE 200000K                                        SP140 SIZE 200000K
                         SP127 SIZE 200000K                                        SP141 SIZE 200000K
                         SP128 SIZE 200000K,                                       SP142 SIZE 200000K
           OVERFLOW ON SP125 SIZE 80001K;                                          SP143 SIZE 200000K
                                                                                   SP144 SIZE 200000K,
CREATE DSI STOCK_32_DSI                                              OVERFLOW ON SP141 SIZE 80001K;
     DSO STOCK_DSO
     USING(1365,1408)                                       CREATE DSI STOCK_37_DSI
     ALLOCATE PRIME    ON SP121 SIZE 200004K                     DSO STOCK_DSO
                         SP122 SIZE 200000K                      USING(1585,1628)
                         SP123 SIZE 200000K                      ALLOCATE PRIME    ON SP145 SIZE 200004K
                         SP124 SIZE 200000K                                        SP146 SIZE 200000K
                         SP125 SIZE 200000K                                        SP147 SIZE 200000K
                         SP126 SIZE 200000K                                        SP148 SIZE 200000K
                         SP127 SIZE 200000K                                        SP149 SIZE 200000K
                         SP128 SIZE 200000K,                                       SP150 SIZE 200000K
           OVERFLOW ON SP125 SIZE 80001K;                                          SP151 SIZE 200000K
                                                                                   SP152 SIZE 200000K,
CREATE DSI STOCK_33_DSI                                              OVERFLOW ON SP149 SIZE 80001K;
     DSO STOCK_DSO
     USING(1409,1452)                                       CREATE DSI STOCK_38_DSI
```

DSO STOCK_DSO
USING(1629,1672)
ALLOCATE PRIME   ON SP145 SIZE 200004K
                    SP146 SIZE 200000K
                    SP147 SIZE 200000K
                    SP148 SIZE 200000K
                    SP149 SIZE 200000K
                    SP150 SIZE 200000K
                    SP151 SIZE 200000K
                    SP152 SIZE 200000K,
        OVERFLOW ON SP149 SIZE 80001K;

CREATE DSI STOCK_39_DSI
    DSO STOCK_DSO
    USING(1673,1716)
    ALLOCATE PRIME   ON SP153 SIZE 200004K
                        SP154 SIZE 200000K
                        SP155 SIZE 200000K
                        SP156 SIZE 200000K
                        SP157 SIZE 200000K
                        SP158 SIZE 200000K
                        SP159 SIZE 200000K
                        SP160 SIZE 200000K,
            OVERFLOW ON SP157 SIZE 80001K;

CREATE DSI STOCK_40_DSI
    DSO STOCK_DSO
    USING(1717,1760)
    ALLOCATE PRIME   ON SP153 SIZE 200004K
                        SP154 SIZE 200000K
                        SP155 SIZE 200000K
                        SP156 SIZE 200000K
                        SP157 SIZE 200000K
                        SP158 SIZE 200000K
                        SP159 SIZE 200000K
                        SP160 SIZE 200000K,
            OVERFLOW ON SP157 SIZE 80001K;

CREATE DSI STOCK_41_DSI
    DSO STOCK_DSO
    USING(1761,1804)
    ALLOCATE PRIME   ON SP161 SIZE 200004K
                        SP162 SIZE 200000K
                        SP163 SIZE 200000K
                        SP164 SIZE 200000K
                        SP165 SIZE 200000K
                        SP166 SIZE 200000K
                        SP167 SIZE 200000K
                        SP168 SIZE 200000K,
            OVERFLOW ON SP165 SIZE 80001K;

CREATE DSI STOCK_42_DSI
    DSO STOCK_DSO
    USING(1805,1848)
    ALLOCATE PRIME   ON SP161 SIZE 200004K
                        SP162 SIZE 200000K
                        SP163 SIZE 200000K
                        SP164 SIZE 200000K
                        SP165 SIZE 200000K
                        SP166 SIZE 200000K
                        SP167 SIZE 200000K
                        SP168 SIZE 200000K,
        OVERFLOW ON SP165 SIZE 80001K;

CREATE DSI STOCK_43_DSI
    DSO STOCK_DSO
    USING(1849,1892)
    ALLOCATE PRIME   ON SP169 SIZE 200004K
                        SP170 SIZE 200000K
                        SP171 SIZE 200000K
                        SP172 SIZE 200000K
                        SP173 SIZE 200000K
                        SP174 SIZE 200000K
                        SP175 SIZE 200000K
                        SP176 SIZE 200000K,
            OVERFLOW ON SP173 SIZE 80001K;

CREATE DSI STOCK_44_DSI
    DSO STOCK_DSO
    USING(1893,1936)
    ALLOCATE PRIME   ON SP169 SIZE 200004K
                        SP170 SIZE 200000K
                        SP171 SIZE 200000K
                        SP172 SIZE 200000K
                        SP173 SIZE 200000K
                        SP174 SIZE 200000K
                        SP175 SIZE 200000K
                        SP176 SIZE 200000K,
            OVERFLOW ON SP173 SIZE 80001K;

CREATE DSI STOCK_45_DSI
    DSO STOCK_DSO
    USING(1937,1980)
    ALLOCATE PRIME   ON SP177 SIZE 200004K
                        SP178 SIZE 200000K
                        SP179 SIZE 200000K
                        SP180 SIZE 200000K
                        SP181 SIZE 200000K
                        SP182 SIZE 200000K
                        SP183 SIZE 200000K
                        SP184 SIZE 200000K,
            OVERFLOW ON SP181 SIZE 80001K;

CREATE DSI STOCK_46_DSI
    DSO STOCK_DSO
    USING(1981,2024)
    ALLOCATE PRIME   ON SP177 SIZE 200004K
                        SP178 SIZE 200000K
                        SP179 SIZE 200000K
                        SP180 SIZE 200000K
                        SP181 SIZE 200000K
                        SP182 SIZE 200000K
                        SP183 SIZE 200000K
                        SP184 SIZE 200000K,
            OVERFLOW ON SP181 SIZE 80001K;

CREATE DSI STOCK_47_DSI
    DSO STOCK_DSO
    USING(2025,2068)
    ALLOCATE PRIME   ON SP185 SIZE 200004K
                        SP186 SIZE 200000K
                        SP187 SIZE 200000K
                        SP188 SIZE 200000K
                        SP189 SIZE 200000K
                        SP190 SIZE 200000K
                        SP191 SIZE 200000K

```
                                  SP192 SIZE 200000K,
                  OVERFLOW ON SP189 SIZE 80001K;

      CREATE DSI STOCK_48_DSI
          DSO STOCK_DSO
          USING(2069,4224)
          ALLOCATE PRIME    ON SP185 SIZE 200004K
                               SP186 SIZE 200000K
                               SP187 SIZE 200000K
                               SP188 SIZE 200000K
                               SP189 SIZE 200000K
                               SP190 SIZE 200000K
                               SP191 SIZE 200000K
                               SP192 SIZE 200000K,
                  OVERFLOW ON SP189 SIZE 80001K;

:::::::::::::
ddl.dat.WH
:::::::::::::
--------------------------------------------------------------------------
-- * Phase.2-1: Warehouse
--------------------------------------------------------------------------
      CREATE DSO WAREHOUSE_DSO
          FROM TPCC_SCHEMA.WAREHOUSE
              TYPE RANDOM(PAGESIZE1(1),PAGESIZE2(1))
              WHERE (W_ID) BETWEEN (?) AND (?);

      CREATE DSI WAREHOUSE_1_DSI
          DSO WAREHOUSE_DSO
           USING(1,88)
           ALLOCATE PRIME    ON SP1 SIZE 4097K,
                  OVERFLOW ON SP1 SIZE 17K;

      CREATE DSI WAREHOUSE_2_DSI
          DSO WAREHOUSE_DSO
           USING(89,176)
           ALLOCATE PRIME    ON SP7 SIZE 4097K,
                  OVERFLOW ON SP7 SIZE 17K;

      CREATE DSI WAREHOUSE_3_DSI
          DSO WAREHOUSE_DSO
           USING(177,264)
           ALLOCATE PRIME    ON SP13 SIZE 4097K,
                  OVERFLOW ON SP13 SIZE 17K;

      CREATE DSI WAREHOUSE_4_DSI
          DSO WAREHOUSE_DSO
           USING(265,352)
           ALLOCATE PRIME    ON SP19 SIZE 4097K,
                  OVERFLOW ON SP19 SIZE 17K;

      CREATE DSI WAREHOUSE_5_DSI
          DSO WAREHOUSE_DSO
           USING(353,440)
           ALLOCATE PRIME    ON SP25 SIZE 4097K,
                  OVERFLOW ON SP25 SIZE 17K;

      CREATE DSI WAREHOUSE_6_DSI
          DSO WAREHOUSE_DSO
           USING(441,528)
           ALLOCATE PRIME    ON SP31 SIZE 4097K,
                  OVERFLOW ON SP31 SIZE 17K;

CREATE DSI WAREHOUSE_7_DSI
    DSO WAREHOUSE_DSO
     USING(529,616)
     ALLOCATE PRIME    ON SP37 SIZE 4097K,
            OVERFLOW ON SP37 SIZE 17K;

CREATE DSI WAREHOUSE_8_DSI
    DSO WAREHOUSE_DSO
     USING(617,704)
     ALLOCATE PRIME    ON SP43 SIZE 4097K,
            OVERFLOW ON SP43 SIZE 17K;

CREATE DSI WAREHOUSE_9_DSI
    DSO WAREHOUSE_DSO
     USING(705,792)
     ALLOCATE PRIME    ON SP49 SIZE 4097K,
            OVERFLOW ON SP49 SIZE 17K;

CREATE DSI WAREHOUSE_10_DSI
    DSO WAREHOUSE_DSO
     USING(793,880)
     ALLOCATE PRIME    ON SP55 SIZE 4097K,
            OVERFLOW ON SP55 SIZE 17K;

CREATE DSI WAREHOUSE_11_DSI
    DSO WAREHOUSE_DSO
     USING(881,968)
     ALLOCATE PRIME    ON SP61 SIZE 4097K,
            OVERFLOW ON SP61 SIZE 17K;

CREATE DSI WAREHOUSE_12_DSI
    DSO WAREHOUSE_DSO
     USING(969,1056)
     ALLOCATE PRIME    ON SP67 SIZE 4097K,
            OVERFLOW ON SP67 SIZE 17K;

CREATE DSI WAREHOUSE_13_DSI
    DSO WAREHOUSE_DSO
     USING(1057,1144)
     ALLOCATE PRIME    ON SP73 SIZE 4097K,
            OVERFLOW ON SP73 SIZE 17K;

CREATE DSI WAREHOUSE_14_DSI
    DSO WAREHOUSE_DSO
     USING(1145,1232)
     ALLOCATE PRIME    ON SP79 SIZE 4097K,
            OVERFLOW ON SP79 SIZE 17K;

CREATE DSI WAREHOUSE_15_DSI
    DSO WAREHOUSE_DSO
     USING(1233,1320)
     ALLOCATE PRIME    ON SP85 SIZE 4097K,
            OVERFLOW ON SP85 SIZE 17K;

CREATE DSI WAREHOUSE_16_DSI
    DSO WAREHOUSE_DSO
     USING(1321,1408)
     ALLOCATE PRIME    ON SP91 SIZE 4097K,
            OVERFLOW ON SP91 SIZE 17K;

CREATE DSI WAREHOUSE_17_DSI
    DSO WAREHOUSE_DSO
     USING(1409,1496)
```

ALLOCATE PRIME   ON SP97 SIZE 4097K,
        OVERFLOW ON SP97 SIZE 17K;

CREATE DSI WAREHOUSE_18_DSI
    DSO WAREHOUSE_DSO
    USING(1497,1584)
        ALLOCATE PRIME   ON SP103 SIZE 4097K,
            OVERFLOW ON SP103 SIZE 17K;

CREATE DSI WAREHOUSE_19_DSI
    DSO WAREHOUSE_DSO
    USING(1585,1672)
        ALLOCATE PRIME   ON SP109 SIZE 4097K,
            OVERFLOW ON SP109 SIZE 17K;

CREATE DSI WAREHOUSE_20_DSI
    DSO WAREHOUSE_DSO
    USING(1673,1760)
        ALLOCATE PRIME   ON SP115 SIZE 4097K,
            OVERFLOW ON SP115 SIZE 17K;

CREATE DSI WAREHOUSE_21_DSI
    DSO WAREHOUSE_DSO
    USING(1761,1848)
        ALLOCATE PRIME   ON SP121 SIZE 4097K,
            OVERFLOW ON SP121 SIZE 17K;

CREATE DSI WAREHOUSE_22_DSI
    DSO WAREHOUSE_DSO
    USING(1849,1936)
        ALLOCATE PRIME   ON SP127 SIZE 4097K,
            OVERFLOW ON SP127 SIZE 17K;

CREATE DSI WAREHOUSE_23_DSI
    DSO WAREHOUSE_DSO
    USING(1937,2024)
        ALLOCATE PRIME   ON SP133 SIZE 4097K,
            OVERFLOW ON SP133 SIZE 17K;

CREATE DSI WAREHOUSE_24_DSI
    DSO WAREHOUSE_DSO
    USING(2025,4224)
        ALLOCATE PRIME   ON SP139 SIZE 4097K,
            OVERFLOW ON SP139 SIZE 17K;

::::::::::::::
ddl.dbsp.192.dat
::::::::::::::
CREATE DBSPACE SP1   ALLOCATE RAWDEVICE
/DEV/rdsk/SP001;
CREATE DBSPACE SP2   ALLOCATE RAWDEVICE
/DEV/rdsk/SP002;
CREATE DBSPACE SP3   ALLOCATE RAWDEVICE
/DEV/rdsk/SP003;
CREATE DBSPACE SP4   ALLOCATE RAWDEVICE
/DEV/rdsk/SP004;
CREATE DBSPACE SP5   ALLOCATE RAWDEVICE
/DEV/rdsk/SP005;
CREATE DBSPACE SP6   ALLOCATE RAWDEVICE
/DEV/rdsk/SP006;
CREATE DBSPACE SP7   ALLOCATE RAWDEVICE
/DEV/rdsk/SP007;

CREATE DBSPACE SP8   ALLOCATE RAWDEVICE /DEV/rdsk/SP008;
CREATE DBSPACE SP9   ALLOCATE RAWDEVICE /DEV/rdsk/SP009;
CREATE DBSPACE SP10 ALLOCATE RAWDEVICE /DEV/rdsk/SP010;
CREATE DBSPACE SP11 ALLOCATE RAWDEVICE /DEV/rdsk/SP011;
CREATE DBSPACE SP12 ALLOCATE RAWDEVICE /DEV/rdsk/SP012;
CREATE DBSPACE SP13 ALLOCATE RAWDEVICE /DEV/rdsk/SP013;
CREATE DBSPACE SP14 ALLOCATE RAWDEVICE /DEV/rdsk/SP014;
CREATE DBSPACE SP15 ALLOCATE RAWDEVICE /DEV/rdsk/SP015;
CREATE DBSPACE SP16 ALLOCATE RAWDEVICE /DEV/rdsk/SP016;
CREATE DBSPACE SP17 ALLOCATE RAWDEVICE /DEV/rdsk/SP017;
CREATE DBSPACE SP18 ALLOCATE RAWDEVICE /DEV/rdsk/SP018;
CREATE DBSPACE SP19 ALLOCATE RAWDEVICE /DEV/rdsk/SP019;
CREATE DBSPACE SP20 ALLOCATE RAWDEVICE /DEV/rdsk/SP020;
CREATE DBSPACE SP21 ALLOCATE RAWDEVICE /DEV/rdsk/SP021;
CREATE DBSPACE SP22 ALLOCATE RAWDEVICE /DEV/rdsk/SP022;
CREATE DBSPACE SP23 ALLOCATE RAWDEVICE /DEV/rdsk/SP023;
CREATE DBSPACE SP24 ALLOCATE RAWDEVICE /DEV/rdsk/SP024;
CREATE DBSPACE SP25 ALLOCATE RAWDEVICE /DEV/rdsk/SP025;
CREATE DBSPACE SP26 ALLOCATE RAWDEVICE /DEV/rdsk/SP026;
CREATE DBSPACE SP27 ALLOCATE RAWDEVICE /DEV/rdsk/SP027;
CREATE DBSPACE SP28 ALLOCATE RAWDEVICE /DEV/rdsk/SP028;
CREATE DBSPACE SP29 ALLOCATE RAWDEVICE /DEV/rdsk/SP029;
CREATE DBSPACE SP30 ALLOCATE RAWDEVICE /DEV/rdsk/SP030;
CREATE DBSPACE SP31 ALLOCATE RAWDEVICE /DEV/rdsk/SP031;
CREATE DBSPACE SP32 ALLOCATE RAWDEVICE /DEV/rdsk/SP032;
CREATE DBSPACE SP33 ALLOCATE RAWDEVICE /DEV/rdsk/SP033;
CREATE DBSPACE SP34 ALLOCATE RAWDEVICE /DEV/rdsk/SP034;
CREATE DBSPACE SP35 ALLOCATE RAWDEVICE /DEV/rdsk/SP035;
CREATE DBSPACE SP36 ALLOCATE RAWDEVICE /DEV/rdsk/SP036;
CREATE DBSPACE SP37 ALLOCATE RAWDEVICE /DEV/rdsk/SP037;
CREATE DBSPACE SP38 ALLOCATE RAWDEVICE /DEV/rdsk/SP038;
CREATE DBSPACE SP39 ALLOCATE RAWDEVICE /DEV/rdsk/SP039;
CREATE DBSPACE SP40 ALLOCATE RAWDEVICE /DEV/rdsk/SP040;
CREATE DBSPACE SP41 ALLOCATE RAWDEVICE /DEV/rdsk/SP041;
CREATE DBSPACE SP42 ALLOCATE RAWDEVICE /DEV/rdsk/SP042;
CREATE DBSPACE SP43 ALLOCATE RAWDEVICE /DEV/rdsk/SP043;
CREATE DBSPACE SP44 ALLOCATE RAWDEVICE /DEV/rdsk/SP044;
CREATE DBSPACE SP45 ALLOCATE RAWDEVICE /DEV/rdsk/SP045;
CREATE DBSPACE SP46 ALLOCATE RAWDEVICE /DEV/rdsk/SP046;
CREATE DBSPACE SP47 ALLOCATE RAWDEVICE /DEV/rdsk/SP047;
CREATE DBSPACE SP48 ALLOCATE RAWDEVICE /DEV/rdsk/SP048;
CREATE DBSPACE SP49 ALLOCATE RAWDEVICE /DEV/rdsk/SP049;
CREATE DBSPACE SP50 ALLOCATE RAWDEVICE /DEV/rdsk/SP050;
CREATE DBSPACE SP51 ALLOCATE RAWDEVICE /DEV/rdsk/SP051;
CREATE DBSPACE SP52 ALLOCATE RAWDEVICE /DEV/rdsk/SP052;
CREATE DBSPACE SP53 ALLOCATE RAWDEVICE /DEV/rdsk/SP053;
CREATE DBSPACE SP54 ALLOCATE RAWDEVICE /DEV/rdsk/SP054;
CREATE DBSPACE SP55 ALLOCATE RAWDEVICE /DEV/rdsk/SP055;
CREATE DBSPACE SP56 ALLOCATE RAWDEVICE /DEV/rdsk/SP056;
CREATE DBSPACE SP57 ALLOCATE RAWDEVICE /DEV/rdsk/SP057;
CREATE DBSPACE SP58 ALLOCATE RAWDEVICE /DEV/rdsk/SP058;
CREATE DBSPACE SP59 ALLOCATE RAWDEVICE /DEV/rdsk/SP059;
CREATE DBSPACE SP60 ALLOCATE RAWDEVICE /DEV/rdsk/SP060;
CREATE DBSPACE SP61 ALLOCATE RAWDEVICE /DEV/rdsk/SP061;
CREATE DBSPACE SP62 ALLOCATE RAWDEVICE /DEV/rdsk/SP062;
CREATE DBSPACE SP63 ALLOCATE RAWDEVICE /DEV/rdsk/SP063;
CREATE DBSPACE SP64 ALLOCATE RAWDEVICE /DEV/rdsk/SP064;
CREATE DBSPACE SP65 ALLOCATE RAWDEVICE /DEV/rdsk/SP065;
CREATE DBSPACE SP66 ALLOCATE RAWDEVICE /DEV/rdsk/SP066;
CREATE DBSPACE SP67 ALLOCATE RAWDEVICE /DEV/rdsk/SP067;
CREATE DBSPACE SP68 ALLOCATE RAWDEVICE /DEV/rdsk/SP068;
CREATE DBSPACE SP69 ALLOCATE RAWDEVICE /DEV/rdsk/SP069;
CREATE DBSPACE SP70 ALLOCATE RAWDEVICE /DEV/rdsk/SP070;

CREATE DBSPACE SP71 ALLOCATE RAWDEVICE /DEV/rdsk/SP071;
CREATE DBSPACE SP72 ALLOCATE RAWDEVICE /DEV/rdsk/SP072;
CREATE DBSPACE SP73 ALLOCATE RAWDEVICE /DEV/rdsk/SP073;
CREATE DBSPACE SP74 ALLOCATE RAWDEVICE /DEV/rdsk/SP074;
CREATE DBSPACE SP75 ALLOCATE RAWDEVICE /DEV/rdsk/SP075;
CREATE DBSPACE SP76 ALLOCATE RAWDEVICE /DEV/rdsk/SP076;
CREATE DBSPACE SP77 ALLOCATE RAWDEVICE /DEV/rdsk/SP077;
CREATE DBSPACE SP78 ALLOCATE RAWDEVICE /DEV/rdsk/SP078;
CREATE DBSPACE SP79 ALLOCATE RAWDEVICE /DEV/rdsk/SP079;
CREATE DBSPACE SP80 ALLOCATE RAWDEVICE /DEV/rdsk/SP080;
CREATE DBSPACE SP81 ALLOCATE RAWDEVICE /DEV/rdsk/SP081;
CREATE DBSPACE SP82 ALLOCATE RAWDEVICE /DEV/rdsk/SP082;
CREATE DBSPACE SP83 ALLOCATE RAWDEVICE /DEV/rdsk/SP083;
CREATE DBSPACE SP84 ALLOCATE RAWDEVICE /DEV/rdsk/SP084;
CREATE DBSPACE SP85 ALLOCATE RAWDEVICE /DEV/rdsk/SP085;
CREATE DBSPACE SP86 ALLOCATE RAWDEVICE /DEV/rdsk/SP086;
CREATE DBSPACE SP87 ALLOCATE RAWDEVICE /DEV/rdsk/SP087;
CREATE DBSPACE SP88 ALLOCATE RAWDEVICE /DEV/rdsk/SP088;
CREATE DBSPACE SP89 ALLOCATE RAWDEVICE /DEV/rdsk/SP089;
CREATE DBSPACE SP90 ALLOCATE RAWDEVICE /DEV/rdsk/SP090;
CREATE DBSPACE SP91 ALLOCATE RAWDEVICE /DEV/rdsk/SP091;
CREATE DBSPACE SP92 ALLOCATE RAWDEVICE /DEV/rdsk/SP092;
CREATE DBSPACE SP93 ALLOCATE RAWDEVICE /DEV/rdsk/SP093;
CREATE DBSPACE SP94 ALLOCATE RAWDEVICE /DEV/rdsk/SP094;
CREATE DBSPACE SP95 ALLOCATE RAWDEVICE /DEV/rdsk/SP095;
CREATE DBSPACE SP96 ALLOCATE RAWDEVICE /DEV/rdsk/SP096;
CREATE DBSPACE SP97 ALLOCATE RAWDEVICE /DEV/rdsk/SP097;
CREATE DBSPACE SP98 ALLOCATE RAWDEVICE /DEV/rdsk/SP098;
CREATE DBSPACE SP99 ALLOCATE RAWDEVICE /DEV/rdsk/SP099;
CREATE DBSPACE SP100 ALLOCATE RAWDEVICE /DEV/rdsk/SP100;
CREATE DBSPACE SP101 ALLOCATE RAWDEVICE /DEV/rdsk/SP101;

CREATE DBSPACE SP102 ALLOCATE RAWDEVICE /DEV/rdsk/SP102;
CREATE DBSPACE SP103 ALLOCATE RAWDEVICE /DEV/rdsk/SP103;
CREATE DBSPACE SP104 ALLOCATE RAWDEVICE /DEV/rdsk/SP104;
CREATE DBSPACE SP105 ALLOCATE RAWDEVICE /DEV/rdsk/SP105;
CREATE DBSPACE SP106 ALLOCATE RAWDEVICE /DEV/rdsk/SP106;
CREATE DBSPACE SP107 ALLOCATE RAWDEVICE /DEV/rdsk/SP107;
CREATE DBSPACE SP108 ALLOCATE RAWDEVICE /DEV/rdsk/SP108;
CREATE DBSPACE SP109 ALLOCATE RAWDEVICE /DEV/rdsk/SP109;
CREATE DBSPACE SP110 ALLOCATE RAWDEVICE /DEV/rdsk/SP110;
CREATE DBSPACE SP111 ALLOCATE RAWDEVICE /DEV/rdsk/SP111;
CREATE DBSPACE SP112 ALLOCATE RAWDEVICE /DEV/rdsk/SP112;
CREATE DBSPACE SP113 ALLOCATE RAWDEVICE /DEV/rdsk/SP113;
CREATE DBSPACE SP114 ALLOCATE RAWDEVICE /DEV/rdsk/SP114;
CREATE DBSPACE SP115 ALLOCATE RAWDEVICE /DEV/rdsk/SP115;
CREATE DBSPACE SP116 ALLOCATE RAWDEVICE /DEV/rdsk/SP116;
CREATE DBSPACE SP117 ALLOCATE RAWDEVICE /DEV/rdsk/SP117;
CREATE DBSPACE SP118 ALLOCATE RAWDEVICE /DEV/rdsk/SP118;
CREATE DBSPACE SP119 ALLOCATE RAWDEVICE /DEV/rdsk/SP119;
CREATE DBSPACE SP120 ALLOCATE RAWDEVICE /DEV/rdsk/SP120;
CREATE DBSPACE SP121 ALLOCATE RAWDEVICE /DEV/rdsk/SP121;
CREATE DBSPACE SP122 ALLOCATE RAWDEVICE /DEV/rdsk/SP122;
CREATE DBSPACE SP123 ALLOCATE RAWDEVICE /DEV/rdsk/SP123;
CREATE DBSPACE SP124 ALLOCATE RAWDEVICE /DEV/rdsk/SP124;
CREATE DBSPACE SP125 ALLOCATE RAWDEVICE /DEV/rdsk/SP125;
CREATE DBSPACE SP126 ALLOCATE RAWDEVICE /DEV/rdsk/SP126;
CREATE DBSPACE SP127 ALLOCATE RAWDEVICE /DEV/rdsk/SP127;
CREATE DBSPACE SP128 ALLOCATE RAWDEVICE /DEV/rdsk/SP128;
CREATE DBSPACE SP129 ALLOCATE RAWDEVICE /DEV/rdsk/SP129;
CREATE DBSPACE SP130 ALLOCATE RAWDEVICE /DEV/rdsk/SP130;
CREATE DBSPACE SP131 ALLOCATE RAWDEVICE /DEV/rdsk/SP131;
CREATE DBSPACE SP132 ALLOCATE RAWDEVICE /DEV/rdsk/SP132;
CREATE DBSPACE SP133 ALLOCATE RAWDEVICE /DEV/rdsk/SP133;
CREATE DBSPACE SP134 ALLOCATE RAWDEVICE /DEV/rdsk/SP134;
CREATE DBSPACE SP135 ALLOCATE RAWDEVICE /DEV/rdsk/SP135;
CREATE DBSPACE SP136 ALLOCATE RAWDEVICE /DEV/rdsk/SP136;
CREATE DBSPACE SP137 ALLOCATE RAWDEVICE /DEV/rdsk/SP137;
CREATE DBSPACE SP138 ALLOCATE RAWDEVICE /DEV/rdsk/SP138;
CREATE DBSPACE SP139 ALLOCATE RAWDEVICE /DEV/rdsk/SP139;
CREATE DBSPACE SP140 ALLOCATE RAWDEVICE /DEV/rdsk/SP140;
CREATE DBSPACE SP141 ALLOCATE RAWDEVICE /DEV/rdsk/SP141;
CREATE DBSPACE SP142 ALLOCATE RAWDEVICE /DEV/rdsk/SP142;
CREATE DBSPACE SP143 ALLOCATE RAWDEVICE /DEV/rdsk/SP143;
CREATE DBSPACE SP144 ALLOCATE RAWDEVICE /DEV/rdsk/SP144;
CREATE DBSPACE SP145 ALLOCATE RAWDEVICE /DEV/rdsk/SP145;
CREATE DBSPACE SP146 ALLOCATE RAWDEVICE /DEV/rdsk/SP146;
CREATE DBSPACE SP147 ALLOCATE RAWDEVICE /DEV/rdsk/SP147;
CREATE DBSPACE SP148 ALLOCATE RAWDEVICE /DEV/rdsk/SP148;
CREATE DBSPACE SP149 ALLOCATE RAWDEVICE /DEV/rdsk/SP149;
CREATE DBSPACE SP150 ALLOCATE RAWDEVICE /DEV/rdsk/SP150;
CREATE DBSPACE SP151 ALLOCATE RAWDEVICE /DEV/rdsk/SP151;
CREATE DBSPACE SP152 ALLOCATE RAWDEVICE /DEV/rdsk/SP152;
CREATE DBSPACE SP153 ALLOCATE RAWDEVICE /DEV/rdsk/SP153;
CREATE DBSPACE SP154 ALLOCATE RAWDEVICE /DEV/rdsk/SP154;
CREATE DBSPACE SP155 ALLOCATE RAWDEVICE /DEV/rdsk/SP155;
CREATE DBSPACE SP156 ALLOCATE RAWDEVICE /DEV/rdsk/SP156;
CREATE DBSPACE SP157 ALLOCATE RAWDEVICE /DEV/rdsk/SP157;
CREATE DBSPACE SP158 ALLOCATE RAWDEVICE /DEV/rdsk/SP158;
CREATE DBSPACE SP159 ALLOCATE RAWDEVICE /DEV/rdsk/SP159;
CREATE DBSPACE SP160 ALLOCATE RAWDEVICE /DEV/rdsk/SP160;
CREATE DBSPACE SP161 ALLOCATE RAWDEVICE /DEV/rdsk/SP161;
CREATE DBSPACE SP162 ALLOCATE RAWDEVICE /DEV/rdsk/SP162;
CREATE DBSPACE SP163 ALLOCATE RAWDEVICE /DEV/rdsk/SP163;
CREATE DBSPACE SP164 ALLOCATE RAWDEVICE /DEV/rdsk/SP164;

```
CREATE DBSPACE SP165  ALLOCATE RAWDEVICE
/DEV/rdsk/SP165;
CREATE DBSPACE SP166  ALLOCATE RAWDEVICE
/DEV/rdsk/SP166;
CREATE DBSPACE SP167  ALLOCATE RAWDEVICE
/DEV/rdsk/SP167;
CREATE DBSPACE SP168  ALLOCATE RAWDEVICE
/DEV/rdsk/SP168;
CREATE DBSPACE SP169  ALLOCATE RAWDEVICE
/DEV/rdsk/SP169;
CREATE DBSPACE SP170  ALLOCATE RAWDEVICE
/DEV/rdsk/SP170;
CREATE DBSPACE SP171  ALLOCATE RAWDEVICE
/DEV/rdsk/SP171;
CREATE DBSPACE SP172  ALLOCATE RAWDEVICE
/DEV/rdsk/SP172;
CREATE DBSPACE SP173  ALLOCATE RAWDEVICE
/DEV/rdsk/SP173;
CREATE DBSPACE SP174  ALLOCATE RAWDEVICE
/DEV/rdsk/SP174;
CREATE DBSPACE SP175  ALLOCATE RAWDEVICE
/DEV/rdsk/SP175;
CREATE DBSPACE SP176  ALLOCATE RAWDEVICE
/DEV/rdsk/SP176;
CREATE DBSPACE SP177  ALLOCATE RAWDEVICE
/DEV/rdsk/SP177;
CREATE DBSPACE SP178  ALLOCATE RAWDEVICE
/DEV/rdsk/SP178;
CREATE DBSPACE SP179  ALLOCATE RAWDEVICE
/DEV/rdsk/SP179;
CREATE DBSPACE SP180  ALLOCATE RAWDEVICE
/DEV/rdsk/SP180;
CREATE DBSPACE SP181  ALLOCATE RAWDEVICE
/DEV/rdsk/SP181;
CREATE DBSPACE SP182  ALLOCATE RAWDEVICE
/DEV/rdsk/SP182;
CREATE DBSPACE SP183  ALLOCATE RAWDEVICE
/DEV/rdsk/SP183;
CREATE DBSPACE SP184  ALLOCATE RAWDEVICE
/DEV/rdsk/SP184;
CREATE DBSPACE SP185  ALLOCATE RAWDEVICE
/DEV/rdsk/SP185;
CREATE DBSPACE SP186  ALLOCATE RAWDEVICE
/DEV/rdsk/SP186;
CREATE DBSPACE SP187  ALLOCATE RAWDEVICE
/DEV/rdsk/SP187;
CREATE DBSPACE SP188  ALLOCATE RAWDEVICE
/DEV/rdsk/SP188;
CREATE DBSPACE SP189  ALLOCATE RAWDEVICE
/DEV/rdsk/SP189;
CREATE DBSPACE SP190  ALLOCATE RAWDEVICE
/DEV/rdsk/SP190;
CREATE DBSPACE SP191  ALLOCATE RAWDEVICE
/DEV/rdsk/SP191;
CREATE DBSPACE SP192  ALLOCATE RAWDEVICE
/DEV/rdsk/SP192;

::::::::::::::
mktmplog.sh
::::::::::::::
:
set -x
date
```

```
echo "its takes 25 min (990424 9GB not duplex)"
rm /rdbptc/tpcc80/SYS/rdblogmanage
timex rdblog -I

LOG_AI=/DEV/rdsk/AI_LOG
LOG_BI=/DEV/rdsk/BI_LOG
LOG_IX=/DEV/rdsk/IX_LOG

rm /rdbptc/arc/ARC* &

#timex rdblog -G -t -c 6600M -io 2048 $LOG_IX $LOG_BI $LOG_AI 4000M 8000M
460
#timex rdblog -G -t -c 5400M -io 2048 $LOG_IX $LOG_BI $LOG_AI 2000M 6000M
200
timex rdblog -G -t -c 5400M -io 2048 $LOG_IX $LOG_BI $LOG_AI 2000M 6000M
300

#timex rdblog -G -t -c 32M -io 2048 $LOG_IX $LOG_BI $LOG_AI 32M 64M 340

wait
::::::::::::::
B.more
::::::::::::::
::::::::::::::
CRDIC.notlog
::::::::::::::
:
set -x
SYS=/rdbptc/tpcc80/SYS
LOG_AI=/DEV/rdsk/AI_LOG
LOG_BI=/DEV/rdsk/BI_LOG
LOG_IX=/DEV/rdsk/IX_LOG
DIC_PL=/DEV/rdsk/DIC

BISZ=64M
AISZ=64M
TRN=100
RCV=32M

date
#rm -f $SYS/rdblogmanage

#rdblog -I
#timex rdblog -G -t -c $RCV -io 2048 $LOG_IX $LOG_BI $LOG_AI $BISZ $AISZ
$TRN
timex rdbcrdic -du 37M -r $DIC_PL
date

rdbstart
rdbstop
::::::::::::::
LOAD.sh.ST
::::::::::::::
#!/bin/csh -xf

setenv RDBDB TPCC

set LOAD1_D = /rdb/loaddata/2
set LOAD2_D = /rdb/loaddata/3
set LOAD3_D = /rdb/loaddata/4
set LOAD4_D = /rdb/loaddata/6
set LOAD5_D = /rdb/loaddata/1
set LOAD6_D = /rdb/loaddata/5
```

```
set LOAD7_D = /rdb/loaddata/7                        wttpccd1 $LOAD1_D $make_s1 $make_e1 S &
set LOAD8_D = /rdb/loaddata/8                        wttpccd1 $LOAD2_D $make_s2 $make_e2 S &
set WK1_D = /rdb/sortwk1                              wttpccd1 $LOAD3_D $make_s3 $make_e3 S &
set WK2_D = /rdb/sortwk2                              wttpccd1 $LOAD4_D $make_s4 $make_e4 S &
set WK3_D = /rdb/sortwk3                              wttpccd1 $LOAD5_D $make_s1b $make_e1b S &
set WK4_D = /rdb/sortwk4                              wttpccd1 $LOAD6_D $make_s2b $make_e2b S &
set WK5_D = /rdb/sortwk5                              wttpccd1 $LOAD7_D $make_s3b $make_e3b S &
set WK6_D = /rdb/sortwk6                              wttpccd1 $LOAD8_D $make_s4b $make_e4b S &
set WK7_D = /rdb/sortwk7
set WK8_D = /rdb/sortwk8                                  wait

rm /rdb/loaddata/*/??*_* /rdb/loaddata/*/data    if ( $num != 0 ) then
rm /rdb/sortwk*/SRT*                                  rm $LOAD1_D/ST$rm_make_s1\_$rm_make_e1
                                                     rm $LOAD2_D/ST$rm_make_s2\_$rm_make_e2
## STOCK                                             rm $LOAD3_D/ST$rm_make_s3\_$rm_make_e3
                                                     rm $LOAD4_D/ST$rm_make_s4\_$rm_make_e4
foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 )         rm $LOAD5_D/ST$rm_make_s1b\_$rm_make_e1b
                                                     rm $LOAD6_D/ST$rm_make_s2b\_$rm_make_e2b
if ( $num != 0 ) then                                rm $LOAD7_D/ST$rm_make_s3b\_$rm_make_e3b
                                                     rm $LOAD8_D/ST$rm_make_s4b\_$rm_make_e4b
        @ rm_make_s1 = $make_s1                   endif
        @ rm_make_e1 = $make_e1
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s3 = $make_s3               if ( $dsi_num1 <= 50 ) then
        @ rm_make_e3 = $make_e3                   timex rdbsloader -mi -i $RDBDB.STOCK_$dsi_num1\_DSI \
        @ rm_make_s4 = $make_s4                                               -s $WK1_D \
        @ rm_make_e4 = $make_e4                                               -s $WK2_D \
                                                     -n $LOAD1_D/ST$make_s1\_$make_e1
        @ rm_make_s1b = $make_s1b            $LOAD5_D/ST$make_s1b\_$make_e1b &
        @ rm_make_e1b = $make_e1b           endif
        @ rm_make_s2b = $make_s2b           if ( $dsi_num2 <= 50 ) then
        @ rm_make_e2b = $make_e2b                timex rdbsloader -mi -i $RDBDB.STOCK_$dsi_num2\_DSI \
        @ rm_make_s3b = $make_s3b                                            -s $WK3_D \
        @ rm_make_e3b = $make_e3b                                            -s $WK4_D \
        @ rm_make_s4b = $make_s4b                        -n $LOAD2_D/ST$make_s2\_$make_e2
        @ rm_make_e4b = $make_e4b           $LOAD6_D/ST$make_s2b\_$make_e2b &
                                            endif
endif                                       if ( $dsi_num3 <= 50 ) then
                                                 timex rdbsloader -mi -i $RDBDB.STOCK_$dsi_num3\_DSI \
        @ make_s1 = $num * 176 + 1                                           -s $WK5_D \
        @ make_e1 = $num * 176 + 44 - 22                                     -s $WK6_D \
        @ make_s2 = $num * 176 + 45                       -n $LOAD3_D/ST$make_s3\_$make_e3
        @ make_e2 = $num * 176 + 88 - 22    $LOAD7_D/ST$make_s3b\_$make_e3b&
        @ make_s3 = $num * 176 + 89         endif
        @ make_e3 = $num * 176 + 132 - 22   if ( $dsi_num4 <= 50 ) then
        @ make_s4 = $num * 176 + 133             timex rdbsloader -mi -i $RDBDB.STOCK_$dsi_num4\_DSI \
        @ make_e4 = $num * 176 + 176 - 22                                    -s $WK7_D \
                                                                            -s $WK8_D \
        @ make_s1b = $num * 176 + 1 + 22                 -n $LOAD4_D/ST$make_s4\_$make_e4
        @ make_e1b = $num * 176 + 44        $LOAD8_D/ST$make_s4b\_$make_e4b &
        @ make_s2b = $num * 176 + 45 + 22   endif
        @ make_e2b = $num * 176 + 88
        @ make_s3b = $num * 176 + 89 + 22
        @ make_e3b = $num * 176 + 132
        @ make_s4b = $num * 176 + 133 + 22  end
        @ make_e4b = $num * 176 + 176
                                                    wait
        @ dsi_num1 = $num * 4 + 1           rm /rdb/loaddata/*/ST*
        @ dsi_num2 = $num * 4 + 2           ::::::::::::::
        @ dsi_num3 = $num * 4 + 3           Y_DELIVERY
        @ dsi_num4 = $num * 4 + 4           ::::::::::::::
                                            -- /*********STORED PROCEDURE*****************************************/
```

```
-- /** Y_DELIVERY   COPYRIGHT FUJITSU LIMITED 1997
**/
-- /**    :                                      **/
-- /**    :                                      **/
-- /**    :SymfoWARE RDB TPC-C Benchmark         **/
-- /**    :  Delivery                            **/
-- /**    :1996/10/12                            **/
-- /**    1997/03/13 Revision3.3 : Any Error(Clause 2.3.6)   **/
-- /*****************************************************************/

-- #RESULT_JOIN          VARCHAR(100)
-- +----------------------------------+
-- | sqllen      short                |
-- +-sqlver---------------------------+  ---
-- | #RESULT_O_IDn     CHAR(9)        |   |
-- +----------------------------------+   |
-- |              CHAR(1) "/"     |   |
-- +----------------------------------+  ---
--
-- +----------------------------------+
--
--
-- ****************************************************************************
EXEC SQL
CREATE PROCEDURE TPCC_SCHEMA.Y_DELIVERY(OUT
#STATE          CHAR(5),
                        INOUT #ERRPOS          INTEGER ,
                        IN   #W_ID            SMALLINT ,
                        INOUT #C_ID            INTEGER ,
                        IN   #O_CARRIER_ID     SMALLINT ,
                        IN   #OL_DELIVERY_D    CHAR(14) ,
                        INOUT #RESULT_JOIN     VARCHAR(100)
                        )

DELIVERY:BEGIN
-- DECLARE

    DECLARE SQLSTATE          CHAR(5)     DEFAULT '00000'; --$$
    DECLARE SAPSTOP           CHAR(1)     DEFAULT '/'   ;  --$
    DECLARE @OL_TOTAL         INTEGER ;                    --$
    DECLARE @DMY_W_ID         SMALLINT;                    --$
    DECLARE @DMY_D_ID         SMALLINT;                    --$
    DECLARE @D_ID          SMALLINT;
    DECLARE @NO_O_ID          INTEGER ;
--   DECLARE @OZAWK          SMALLINT;


-- (3) ORDERS   table cursor
    DECLARE CDOS CURSOR FOR
        SELECT O_C_ID
            FROM  TPCC_SCHEMA.ORDERS
            WHERE O_W_ID = #W_ID
              AND O_D_ID = @D_ID
              AND O_ID   = @NO_O_ID
            FOR UPDATE;

--   SET  @OZAWK = 1;

-- LOOP
   SET  @D_ID = 1;
   DID10:LOOP
      IF  @D_ID > 10 THEN
```

```
   GOTO NORMAL_END ;
END IF;
-- (1) NEWORDER                    NO_O_ID
--
                        WHENEVER SQLERROR  GOTO ERR_S_NO;
SELECT MIN( NO_O_ID )
    INTO  @NO_O_ID
    FROM  TPCC_SCHEMA.NEWORDER
    WHERE NO_W_ID = #W_ID
      AND NO_D_ID = @D_ID;
                        WHENEVER SQLERROR  CONTINUE;
IF  SQLSTATE <> '00000'
  OR @NO_O_ID IS NULL   THEN
    SET  @NO_O_ID  = 99999999   ;
    GOTO  NEXT_DID ;
END IF;

-- (2) NEW-ORDER          @NO_O_ID
--
                        WHENEVER SQLERROR  GOTO ERR_D_NO;
                        WHENEVER NOT FOUND GOTO ERR_D_NO;
DELETE FROM TPCC_SCHEMA.NEWORDER
    WHERE NO_W_ID = #W_ID
      AND NO_D_ID = @D_ID
      AND NO_O_ID = @NO_O_ID ;

-- (5) ORDER-LINE          OL_AMOUNT
                        WHENEVER SQLERROR  GOTO ERR_S_OL;
                        WHENEVER NOT FOUND GOTO ERR_S_OL;
SELECT SUM(OL_AMOUNT)
    INTO  @OL_TOTAL
    FROM  TPCC_SCHEMA.ORDERLINE
        WHERE OL_W_ID = #W_ID
          AND OL_D_ID = @D_ID
          AND OL_O_ID = @NO_O_ID
          AND OL_NUMBER IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);
--$$              AND OL_NUMBER = @OZAWK;
--$ -- if OL index exist
--$        WHERE OL_W_ID = #W_ID
--$          AND OL_D_ID = @D_ID
--$          AND OL_O_ID = @NO_O_ID;
      -- ORDER-LINE
                        WHENEVER SQLERROR  GOTO ERR_U_OL;
                        WHENEVER NOT FOUND GOTO ERR_U_OL;
UPDATE      TPCC_SCHEMA.ORDERLINE
    SET   OL_DELIVERY_D = #OL_DELIVERY_D
      WHERE   OL_W_ID = #W_ID
          AND OL_D_ID = @D_ID
          AND OL_O_ID = @NO_O_ID
          AND OL_NUMBER IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);
--$ -- if OL index exist
--$        WHERE OL_W_ID = #W_ID
--$          AND OL_D_ID = @D_ID
--$          AND OL_O_ID = @NO_O_ID;

-- (3) ORDER          @NO_O_ID
--
                        WHENEVER SQLERROR  GOTO ERR_S_OR;
                        WHENEVER NOT FOUND GOTO ERR_S_OR;
OPEN   CDOS;
```

```
    FETCH  CDOS INTO #C_ID;                              SET #ERRPOS = 307    ;
                    WHENEVER SQLERROR  CONTINUE;         SET #STATE  = SQLSTATE;
                    WHENEVER NOT FOUND CONTINUE;         ROLLBACK WORK        ;
                                                         LEAVE DELIVERY       ;
    -- (4) ORDER                                    ERR_U_OL:
                    WHENEVER SQLERROR  GOTO ERR_U_OR;    SET #ERRPOS = 308    ;
    UPDATE     TPCC_SCHEMA.ORDERS                        SET #STATE  = SQLSTATE;
        SET   O_CARRIER_ID = #O_CARRIER_ID               ROLLBACK WORK        ;
        WHERE CURRENT OF CDOS;                           LEAVE DELIVERY       ;
                    WHENEVER SQLERROR  CONTINUE;    ERR_D_NO:
    CLOSE  CDOS;                                         SET #ERRPOS = 409    ;
                                                         SET #STATE  = SQLSTATE;
    -- (6)                  Customer                     ROLLBACK WORK        ;
                    WHENEVER SQLERROR  GOTO
ERR_U_CM;                                         END DELIVERY
                    WHENEVER NOT FOUND GOTO        END-EXEC;
ERR_U_CM;
    UPDATE     TPCC_SCHEMA.CUSTOMER              ::::::::::::::
        SET   C_BALANCE     = C_BALANCE + @OL_TOTAL,  Y_NORDER
           C_DELIVERY_CNT = C_DELIVERY_CNT + 1     ::::::::::::::
           WHERE C_W_ID  = #W_ID                  -- /*********STORED PROCEDURE***************************************/
            AND C_D_ID  = @D_ID                   -- /** Y_NORDER     COPYRIGHT FUJITSU LIMITED 1997        **/
            AND C_ID    = #C_ID;                  -- /**   :                                              **/
                    WHENEVER SQLERROR  CONTINUE;  -- /**   :                                              **/
                    WHENEVER NOT FOUND CONTINUE;  -- /**   :SymfoWARE RDB TPC-C Benchmark                 **/
NEXT_DID:                                         -- /**   :  NewOrder                                    **/
                                                  -- /**   :1996/10/12                            **/
    SET  #RESULT_JOIN = #RESULT_JOIN              -- /**     1997/03/13  Revision3.3 : Any Error(Clause 2.3.6)    **/
            || CAST(@NO_O_ID AS CHAR(9)) || SAPSTOP ;  -- /**************************************************************/

    SET  @D_ID = @D_ID + 1;                       -- *********************************************************************
                                                  -- #S_JOIN              VARCHAR(1215)
    COMMIT WORK          ;                         -- +-----------------------------------+
                                                  -- | sqllen      short       |
  END LOOP DID10;                                 -- +-sqlver----------------------------+  ---
-- LOOP END                                       -- | S_QUANTITYn    CHAR(6)          | |
                                                  -- +-----------------------------------+  |
NORMAL_END:                                       -- | S_DISTn        CHAR(24)         | |
    SET #STATE = '00000' ;                        -- +-----------------------------------+  |
    LEAVE DELIVERY      ;                         -- | S_DATAn        CHAR(50)         | |
                                                  -- +-----------------------------------+  |
--SQLERR:NOT_OUND:                                -- |              CHAR(1) "/"     | |
ERR_S_OR:                                         -- +-----------------------------------+  ---
    SET #ERRPOS = 207    ;                        -- |                   |
    SET #STATE  = SQLSTATE;                       --
    ROLLBACK WORK        ;                        -- +-----------------------------------+
    LEAVE DELIVERY       ;                        --
ERR_S_OL:                                         --
    SET #ERRPOS = 208    ;                        --
    SET #STATE  = SQLSTATE;                       -- #I_JOIN              VARCHAR(1215)
    ROLLBACK WORK        ;                        -- +-----------------------------------+
    LEAVE DELIVERY       ;                        -- | sqllen      short       |
ERR_S_NO:                                         -- +-sqlver----------------------------+  ---
    SET #ERRPOS = 209    ;                        -- | I_PRICEHn      CHAR(6)          | |
    SET #STATE  = SQLSTATE;                       -- +-----------------------------------+  |
    ROLLBACK WORK        ;                        -- | I_NAMEn        CHAR(24)         | |
    LEAVE DELIVERY       ;                        -- +-----------------------------------+  |
ERR_U_CM:                                         -- | I_DATAn        CHAR(50)         | |
    SET #ERRPOS = 305    ;                        -- +-----------------------------------+  |
    SET #STATE  = SQLSTATE;                       -- |              CHAR(1) "/"     | |
    ROLLBACK WORK        ;                        -- +-----------------------------------+  ---
    LEAVE DELIVERY       ;                        -- |                   |
ERR_U_OR:                                         --
```

```
-- +------------------------------------+
--
--
-- ****************************************************************************
EXEC SQL
CREATEPROCEDURETPCC_SCHEMA.Y_NORDER(OUT #STATE
CHAR(5),
                    INOUT #ERRPOS         INTEGER ,
                    IN   #W_ID            SMALLINT,
                    IN   #D_ID            SMALLINT,
                    IN   #C_ID            INTEGER ,
                    INOUT #O_ALL_LOCAL    SMALLINT,
                    OUT  #W_TAX           SMALLINT,
                    OUT  #D_TAX           SMALLINT,
                    INOUT #O_ID           INTEGER ,
                    IN   #O_ENTRY_D       CHAR(14),
                    OUT  #C_DISCOUNT      SMALLINT,
                    OUT  #C_LAST          CHAR(16),
                    OUT  #C_CREDIT        CHAR(2),
                    INOUT #ITEM_NF_CTR    SMALLINT,
                    IN   #H_CNT           SMALLINT,
                    IN   #R_CNT           SMALLINT,
                    IN   #OL_I_ID1        INTEGER,
                    IN   #OL_I_ID2        INTEGER,
                    IN   #OL_I_ID3        INTEGER,
                    IN   #OL_I_ID4        INTEGER,
                    IN   #OL_I_ID5        INTEGER,
                    IN   #OL_I_ID6        INTEGER,
                    IN   #OL_I_ID7        INTEGER,
                    IN   #OL_I_ID8        INTEGER,
                    IN   #OL_I_ID9        INTEGER,
                    IN   #OL_I_ID10       INTEGER,
                    IN   #OL_I_ID11       INTEGER,
                    IN   #OL_I_ID12       INTEGER,
                    IN   #OL_I_ID13       INTEGER,
                    IN   #OL_I_ID14       INTEGER,
                    IN   #OL_I_ID15       INTEGER,
                    IN   #OL_I_ID_JOIN    VARCHAR(105) ,
                IN   #OL_QUANTITY_JOIN VARCHAR(60)
,
                    INOUT #S_JOIN         VARCHAR(1215),
                    INOUT #I_JOIN         VARCHAR(1215),
                IN   #OL_SUPPLY_W_JOIN VARCHAR(60)
                    )
NEWORDER:BEGIN
-- DECLARE
    DECLARE SQLSTATE        CHAR(5)       DEFAULT '00000';
    DECLARE SAPSTOP         CHAR(1)       DEFAULT '/';
    DECLARE @OL_I_ID        INTEGER;
    DECLARE @OL_SUPPLY_W_ID  SMALLINT;
    DECLARE @OL_QUANTITY     SMALLINT;
    DECLARE @S_QUANTITY      SMALLINT;
    DECLARE @I_PRICEH        SMALLINT;
    DECLARE @I_NAMEH         CHAR(24);
    DECLARE @I_DATAH         CHAR(50);
    DECLARE @S_DATA          CHAR(50);
    DECLARE @S_YTD           INTEGER;
    DECLARE @S_ORDER_CNT     SMALLINT;
    DECLARE @S_REMOTE_CNT    SMALLINT;
    DECLARE @D_NEXT_O_ID     INTEGER;
    DECLARE @OL_NUMBER       SMALLINT;
    DECLARE @STOCK_NUM       SMALLINT;
    DECLARE @MATCH_TBL_CNT   SMALLINT;

    DECLARE @S_DIST          CHAR(24);
    DECLARE @S_DIST_01       CHAR(24);
    DECLARE @S_DIST_02       CHAR(24);
    DECLARE @S_DIST_03       CHAR(24);
    DECLARE @S_DIST_04       CHAR(24);
    DECLARE @S_DIST_05       CHAR(24);
    DECLARE @S_DIST_06       CHAR(24);
    DECLARE @S_DIST_07       CHAR(24);
    DECLARE @S_DIST_08       CHAR(24);
    DECLARE @S_DIST_09       CHAR(24);
    DECLARE @S_DIST_10       CHAR(24);
    DECLARE @S_DIST_JOIN     CHAR(240)  ;
    DECLARE @C_OL_I_ID       CHAR(7)    ;
    DECLARE @C_I_PRICEH      CHAR(6)    ;
    DECLARE @C_S_QUANTITY    CHAR(6)    ;
    DECLARE @OL_AMOUNT       INTEGER    ;
    DECLARE @O_OL_CNT        SMALLINT   ;
    DECLARE @DIST_POS        SMALLINT   ;

-- (7) ITEM     table sele(IN)
    DECLARE ITEM_H CURSOR FOR
        SELECT I_PRICE,
               I_NAME,
               I_DATA,
               I_ID
        FROM   TPCC_SCHEMA.ITEM
        WHERE  TPCC_SCHEMA.ITEM.I_ID
               IN( #OL_I_ID1    ,
                   #OL_I_ID2    ,
                   #OL_I_ID3    ,
                   #OL_I_ID4    ,
                   #OL_I_ID5    ,
                   #OL_I_ID6    ,
                   #OL_I_ID7    ,
                   #OL_I_ID8    ,
                   #OL_I_ID9    ,
                   #OL_I_ID10   ,
                   #OL_I_ID11   ,
                   #OL_I_ID12   ,
                   #OL_I_ID13   ,
                   #OL_I_ID14   ,
                   #OL_I_ID15   ) ;
-- (8) STOCK     table select
    DECLARE CNSS_HOME CURSOR FOR
        SELECT S_I_ID,S_QUANTITY,
            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_YTD,S_ORDER_CNT,S_REMOTE_CNT,S_DATA
        FROM  TPCC_SCHEMA.STOCK
        WHERE S_W_ID = #W_ID
          AND S_I_ID  IN( #OL_I_ID1    ,
                   #OL_I_ID2    ,
                   #OL_I_ID3    ,
                   #OL_I_ID4    ,
                   #OL_I_ID5    ,
                   #OL_I_ID6    ,
                   #OL_I_ID7    ,
                   #OL_I_ID8    ,
                   #OL_I_ID9    ,
                   #OL_I_ID10   ,
                   #OL_I_ID11   ,
                   #OL_I_ID12   ,
```

```
                  #OL_I_ID13   ,                                CLOSE  ITEM_H ;
                  #OL_I_ID14   ,
                  #OL_I_ID15   )                     --     (8) STOCK     table select
       ORDER BY S_I_ID                               --     (9) STOCK     table update
          FOR UPDATE ;                                                WHENEVER SQLERROR  GOTO ERR_S_ST;
                                                                      WHENEVER NOT FOUND GOTO ERR_S_ST;
   SET @DIST_POS    = 1+((#D_ID-1)*24);              OPEN  CNSS_HOME  ;
   SET @O_OL_CNT    = #H_CNT + #R_CNT ;                              WHENEVER SQLERROR  CONTINUE;
   SET #O_ALL_LOCAL = 1            ;                                 WHENEVER NOT FOUND CONTINUE;

   -- (4) CUSTOMER  table select                     -- LOOP
                  WHENEVER SQLERROR  GOTO ERR_S_CM;     SET  @STOCK_NUM = 0;
                  WHENEVER NOT FOUND GOTO             OLCNT:LOOP
ERR_S_CM;                                              IF @STOCK_NUM = #H_CNT THEN
   SELECT C_LAST,C_CREDIT,C_DISCOUNT                       GOTO L3 ;
       INTO  #C_LAST,                                  END IF;
             #C_CREDIT,
             #C_DISCOUNT                                 -- (8) STOCK     table select
       FROM  TPCC_SCHEMA.CUSTOMER                                   WHENEVER SQLERROR  GOTO ERR_S_ST;
       WHERE C_W_ID = #W_ID                                        WHENEVER NOT FOUND GOTO L3     ;
         AND C_D_ID = #D_ID                           FETCH  CNSS_HOME
         AND C_ID   = #C_ID;                             INTO @OL_I_ID,@S_QUANTITY,
                  WHENEVER SQLERROR  CONTINUE;            @S_DIST_01,@S_DIST_02,@S_DIST_03,@S_DIST_04,@S_DIST_05,
                  WHENEVER NOT FOUND CONTINUE;           @S_DIST_06,@S_DIST_07,@S_DIST_08,@S_DIST_09,@S_DIST_10,
   IF  #H_CNT = 0 THEN                                   @S_YTD,@S_ORDER_CNT,@S_REMOTE_CNT,@S_DATA;
    GOTO REMORT_PROC ;                                            WHENEVER SQLERROR  CONTINUE;
   END IF;                                                        WHENEVER NOT FOUND CONTINUE;
                                                      SET  @S_DIST_JOIN = @S_DIST_01
HOME_PROC:                                                         || @S_DIST_02
-- Home Warehouse PROCESS START                                    || @S_DIST_03
-- (              Warehouse id            )                        || @S_DIST_04
   -- (7) ITEM      table select                                   || @S_DIST_05
                  WHENEVER SQLERROR  GOTO ERR_S_IT;                || @S_DIST_06
                  WHENEVER NOT FOUND GOTO ERR_S_IT;                || @S_DIST_07
   OPEN  ITEM_H ;                                                  || @S_DIST_08
                  WHENEVER SQLERROR  CONTINUE;                     || @S_DIST_09
                  WHENEVER NOT FOUND CONTINUE;                     || @S_DIST_10                    ;
-- LOOP                                                SET @S_DIST = SUBSTRING(@S_DIST_JOIN FROM @DIST_POS FOR 24)
   SET  @MATCH_TBL_CNT = 0 ;                      ;
   INCNT:LOOP                                          SET @OL_QUANTITY  = CAST(SUBSTRING(#OL_QUANTITY_JOIN
                  WHENEVER SQLERROR  GOTO ERR_S_IT;              FROM 1+(@STOCK_NUM * 4) FOR 4)
                  WHENEVER NOT FOUND GOTO L1;                    AS SMALLINT )                ;
      FETCH  ITEM_H
          INTO @I_PRICEH,                            SET @S_QUANTITY = (@S_QUANTITY - @OL_QUANTITY);
              @I_NAMEH,                              IF  @S_QUANTITY < 10 THEN
              @I_DATAH,                                SET @S_QUANTITY = @S_QUANTITY + 91  ;
              @OL_I_ID;                              END IF;
                  WHENEVER SQLERROR  CONTINUE;
                  WHENEVER NOT FOUND CONTINUE;        SET @S_YTD      = @S_YTD + @OL_QUANTITY;
                                                      SET @S_ORDER_CNT = @S_ORDER_CNT + 1;
      SET  @MATCH_TBL_CNT = @MATCH_TBL_CNT + 1;
                                                        -- (9) STOCK     table update
      SET  @C_I_PRICEH   = CAST(@I_PRICEH AS CHAR(6) )               WHENEVER SQLERROR  GOTO ERR_U_ST;
;                                                     UPDATE TPCC_SCHEMA.STOCK
      SET   #I_JOIN       = #I_JOIN    || @C_I_PRICEH ||    SET  S_QUANTITY   = @S_QUANTITY,
                  @I_NAMEH   || @I_DATAH    || SAPSTOP ;         S_YTD        = @S_YTD,
                                                                S_ORDER_CNT  = @S_ORDER_CNT,
   END LOOP INCNT;                                               S_REMOTE_CNT = @S_REMOTE_CNT
-- LOOP END                                               WHERE CURRENT OF CNSS_HOME ;
                                                                 WHENEVER SQLERROR  CONTINUE;
L1:  IF @MATCH_TBL_CNT < #H_CNT  THEN
       SET   #ITEM_NF_CTR = @MATCH_TBL_CNT ;            SET  @C_S_QUANTITY  = CAST(@S_QUANTITY AS CHAR(6) )        ;
     END IF;
```

```
         SET  #S_JOIN      = #S_JOIN   || @C_S_QUANTITY ||                        S_DIST_01,S_DIST_02,S_DIST_03,S_DIST_04,S_DIST_05,
                  @S_DIST   || @S_DATA      || SAPSTOP ;                          S_DIST_06,S_DIST_07,S_DIST_08,S_DIST_09,S_DIST_10,
                                                                                 S_YTD,S_ORDER_CNT,S_REMOTE_CNT,S_DATA
         SET @STOCK_NUM = @STOCK_NUM + 1;                              INTO @S_QUANTITY,
                                                                          @S_DIST_01,@S_DIST_02,@S_DIST_03,@S_DIST_04,@S_DIST_05,
      END LOOP OLCNT;                                                      @S_DIST_06,@S_DIST_07,@S_DIST_08,@S_DIST_09,@S_DIST_10,
-- LOOP END                                                               @S_YTD,@S_ORDER_CNT,@S_REMOTE_CNT,@S_DATA
                                                                      FROM  TPCC_SCHEMA.STOCK
L3:  IF   @STOCK_NUM <> #H_CNT                                         WHERE S_W_ID = @OL_SUPPLY_W_ID
      AND @STOCK_NUM <> #ITEM_NF_CTR THEN                                AND  S_I_ID = @OL_I_ID  ;
        GOTO ERR_S_ST_NF;                                                          WHENEVER SQLERROR  CONTINUE;
      END IF;                                                                      WHENEVER NOT FOUND CONTINUE;
      CLOSE  CNSS_HOME   ;
                                                                        SET  @S_DIST_JOIN = @S_DIST_01
-- Home Warehouse PROCESS END                                                      || @S_DIST_02
                                                                                   || @S_DIST_03
      IF  #R_CNT = 0 THEN                                                           || @S_DIST_04
        GOTO DISTRICT_PROC ;                                                        || @S_DIST_05
      END IF;                                                                       || @S_DIST_06
                                                                                   || @S_DIST_07
REMORT_PROC:                                                                        || @S_DIST_08
-- Remote Warehouse process start                                                  || @S_DIST_09
-- (                        Warehouse)                                              || @S_DIST_10                      ;
-- LOOP                                                                SET @S_DIST = SUBSTRING(@S_DIST_JOIN FROM @DIST_POS FOR 24)
      SET  @MATCH_TBL_CNT = 0 ;                            ;
      SET  @STOCK_NUM    = 0 ;
                                                                        SET  @OL_QUANTITY  = CAST(SUBSTRING(#OL_QUANTITY_JOIN
      OLCNT_R:LOOP                                                                FROM 1+((@STOCK_NUM+#H_CNT) * 4) FOR 4)
R1:    IF @STOCK_NUM = #R_CNT THEN                                                 AS SMALLINT )                ;
        GOTO R3 ;
      END IF;                                                           SET @S_QUANTITY = (@S_QUANTITY - @OL_QUANTITY);
                                                                       IF  @S_QUANTITY < 10 THEN
      SET  @OL_I_ID      = CAST(SUBSTRING(#OL_I_ID_JOIN                  SET @S_QUANTITY = @S_QUANTITY + 91  ;
                     FROM 1+(@STOCK_NUM * 7) FOR 7)                    END IF;
                  AS INTEGER )                 ;
                                                                       SET @S_YTD      = @S_YTD + @OL_QUANTITY;
      SET  @OL_SUPPLY_W_ID =                                           SET @S_ORDER_CNT = @S_ORDER_CNT + 1;
CAST(SUBSTRING(#OL_SUPPLY_W_JOIN                                       IF  @OL_SUPPLY_W_ID <> #W_ID THEN
                     FROM 1+(@STOCK_NUM * 4) FOR 4)                      SET @S_REMOTE_CNT = @S_REMOTE_CNT + 1;
                  AS SMALLINT )                 ;                        SET #O_ALL_LOCAL  = 0;
   -- (7) ITEM     table select                                       END IF;
                   WHENEVER SQLERROR  GOTO ERR_S_IT ;
                   WHENEVER NOT FOUND GOTO R2      ;                 -- (9) STOCK    table update
   SELECT I_PRICE,I_NAME,I_DATA                                                    WHENEVER SQLERROR  GOTO ERR_U_ST;
       INTO  @I_PRICEH,                                               UPDATE TPCC_SCHEMA.STOCK
           @I_NAMEH ,                                                   SET   S_QUANTITY  = @S_QUANTITY,
           @I_DATAH                                                       S_YTD       = @S_YTD,
       FROM   TPCC_SCHEMA.ITEM                                            S_ORDER_CNT  = @S_ORDER_CNT,
       WHERE  I_ID = @OL_I_ID ;                                          S_REMOTE_CNT = @S_REMOTE_CNT
                   WHENEVER SQLERROR  CONTINUE;                       WHERE   S_W_ID = @OL_SUPPLY_W_ID
                   WHENEVER NOT FOUND CONTINUE;                         AND   S_I_ID = @OL_I_ID  ;
                                                                                   WHENEVER SQLERROR  CONTINUE;
      SET   @MATCH_TBL_CNT = @MATCH_TBL_CNT + 1    ;
                                                                      SET  @C_S_QUANTITY  = CAST(@S_QUANTITY AS CHAR(6) )        ;
      SET  @C_I_PRICEH   = CAST(@I_PRICEH AS CHAR(6) )       ;       SET  #S_JOIN      = #S_JOIN    || @C_S_QUANTITY ||
      SET  #I_JOIN       = #I_JOIN    || @C_I_PRICEH ||                        @S_DIST   || @S_DATA      || SAPSTOP ;
                  @I_NAMEH  || @I_DATAH   || SAPSTOP ;
                                                                   R2:   SET @STOCK_NUM = @STOCK_NUM + 1 ;
      -- (8) STOCK    table select
                   WHENEVER SQLERROR  GOTO ERR_S_ST;                   END LOOP OLCNT_R;
                   WHENEVER NOT FOUND GOTO
ERR_S_ST;                                                          -- LOOP END
      SELECT  S_QUANTITY,                                          R3:
```

```
    IF  @MATCH_TBL_CNT < #R_CNT  THEN        NORMAL_END:
      IF #ITEM_NF_CTR = -1 THEN                  SET #STATE = '00000' ;
        SET   #ITEM_NF_CTR = @MATCH_TBL_CNT ;    LEAVE NEWORDER ;
      ELSE
        SET   #ITEM_NF_CTR = #ITEM_NF_CTR +  --SQLERR:NOT_FOUND:
@MATCH_TBL_CNT ;                           ERR_I_OR:
      END IF;                                    SET #ERRPOS = 107    ;
    END IF;                                      SET #STATE  = SQLSTATE;
                                                 LEAVE NEWORDER       ;
-- Remote Warehouse process end             ERR_I_OL:
                                                 SET #ERRPOS = 108    ;
DISTRICT_PROC:                                   SET #STATE  = SQLSTATE;
    -- (3) DISTRICT  table update                LEAVE NEWORDER       ;
                  WHENEVER SQLERROR  GOTO ERR_U_DI;  ERR_I_NO:
                  WHENEVER NOT FOUND GOTO           SET #ERRPOS = 109    ;
ERR_U_DI;                                          SET #STATE  = SQLSTATE;
    UPDATE TPCC_SCHEMA.DISTRICT                    LEAVE NEWORDER       ;
        SET   D_NEXT_O_ID = D_NEXT_O_ID+1      ERR_S_IT:
        WHERE D_W_ID     = #W_ID                  SET #ERRPOS = 201    ;
          AND D_ID       = #D_ID ;                SET #STATE  = SQLSTATE;
                  WHENEVER SQLERROR  CONTINUE;     LEAVE NEWORDER       ;
                  WHENEVER NOT FOUND CONTINUE;  ERR_S_WH:
    -- (2) DISTRICT  table select                 SET #ERRPOS = 202    ;
                                                  SET #STATE  = SQLSTATE;
    SELECT  D_NEXT_O_ID-1,D_TAX                   LEAVE NEWORDER       ;
        INTO  #O_ID ,#D_TAX                     ERR_S_DI:
        FROM  TPCC_SCHEMA.DISTRICT                SET #ERRPOS = 203    ;
        WHERE D_W_ID = #W_ID                      SET #STATE  = SQLSTATE;
          AND D_ID   = #D_ID ;                    LEAVE NEWORDER       ;
                                               ERR_S_ST:
    -- (6) ORDERS    table insert                 SET #ERRPOS = 204    ;
                  WHENEVER SQLERROR  GOTO ERR_I_OR;  SET #STATE  = SQLSTATE;
                  WHENEVER NOT FOUND GOTO           LEAVE NEWORDER       ;
ERR_I_OR;                                       ERR_S_ST_NF:
    INSERT INTO TPCC_SCHEMA.ORDERS                SET #ERRPOS = 204    ;
        VALUES (#O_ID,                            SET #STATE  = '02000' ;
            #D_ID,                                LEAVE NEWORDER       ;
            #W_ID,                             ERR_S_CM:
            #C_ID,                                SET #ERRPOS = 205    ;
            #O_ENTRY_D,                           SET #STATE  = SQLSTATE;
            NULL,                                 LEAVE NEWORDER       ;
            @O_OL_CNT,                         ERR_U_DI:
            #O_ALL_LOCAL);                        SET #ERRPOS = 303    ;
                  WHENEVER SQLERROR  CONTINUE;     SET #STATE  = SQLSTATE;
                  WHENEVER NOT FOUND CONTINUE;     LEAVE NEWORDER       ;
    -- (5) NEWORDER  table insert              ERR_U_ST:
                  WHENEVER SQLERROR  GOTO ERR_I_NO;  SET #ERRPOS = 304    ;
                  WHENEVER NOT FOUND GOTO           SET #STATE  = SQLSTATE;
ERR_I_NO;
    INSERT INTO TPCC_SCHEMA.NEWORDER           END NEWORDER
        VALUES (#O_ID,                         END-EXEC;
            #D_ID,                             :::::::::::::
            #W_ID);                            Y_NORDER5
                  WHENEVER SQLERROR  CONTINUE;  :::::::::::::
                  WHENEVER NOT FOUND CONTINUE; -- /*********STORED PROCEDURE*******************************************/
    -- (1) WAREHOUSE table update             -- /** Y_NORDER    COPYRIGHT FUJITSU LIMITED 1997          **/
                  WHENEVER SQLERROR  GOTO      -- /**    :                                    **/
ERR_S_WH;                                      -- /**    :                                    **/
    SELECT W_TAX                               -- /**    :SymfoWARE RDB TPC-C Benchmark             **/
        INTO  #W_TAX                           -- /**    :   NewOrder                         **/
        FROM  TPCC_SCHEMA.WAREHOUSE            -- /**    :1996/10/12                          **/
        WHERE W_ID=#W_ID ;                     -- /**    1997/03/13 Revision3.3 : Any Error(Clause 2.3.6)    **/
                  WHENEVER SQLERROR  CONTINUE;
```

```
-- /**     1999/05/27          S_JOIN,I_JOIN        **/
-- /****************************************************************/

-- ***********************************************************************
-- #S_JOIN            VARCHAR(1215)
-- +-----------------------------------+
-- | sqllen       short          |
-- +-sqlver----------------------------+ ---
-- | S_QUANTITYn    CHAR(6)        |  |
-- +-----------------------------------+  |
-- | S_DISTn        CHAR(24)       |  |
-- +-----------------------------------+  |
-- | S_DATAn        CHAR(50)       |  |
-- +-----------------------------------+  |
-- |              CHAR(1) "/"      |  |
-- +-----------------------------------+ ---
-- |                    |
--
-- +-----------------------------------+
--
--
--
-- #I_JOIN            VARCHAR(1215)
-- +-----------------------------------+
-- | sqllen       short          |
-- +-sqlver----------------------------+ ---
-- | I_PRICEHn      CHAR(6)        |  |
-- +-----------------------------------+  |
-- | I_NAMEn        CHAR(24)       |  |
-- +-----------------------------------+  |
-- | I_DATAn        CHAR(50)       |  |
-- +-----------------------------------+  |
-- |              CHAR(1) "/"      |  |
-- +-----------------------------------+ ---
-- |                    |
--
-- +-----------------------------------+
--
--
-- ***********************************************************************
EXEC SQL
CREATE PROCEDURE TPCC_SCHEMA.Y_NORDER5(OUT
#STATE        CHAR(5),
                INOUT #ERRPOS        INTEGER ,
                IN   #W_ID         SMALLINT,
                IN   #D_ID         SMALLINT,
                IN   #C_ID         INTEGER ,
                INOUT #O_ALL_LOCAL    SMALLINT,
                OUT  #W_TAX        SMALLINT,
                OUT  #D_TAX        SMALLINT,
                INOUT #O_ID        INTEGER ,
                IN   #O_ENTRY_D     CHAR(14),
                OUT  #C_DISCOUNT     SMALLINT,
                OUT  #C_LAST        CHAR(16),
                OUT  #C_CREDIT       CHAR(2),
                INOUT #ITEM_NF_CTR     SMALLINT,
                IN   #H_CNT        SMALLINT,
                IN   #R_CNT        SMALLINT,
                IN   #OL_I_ID1       INTEGER,
                IN   #OL_I_ID2       INTEGER,
                IN   #OL_I_ID3       INTEGER,
                IN   #OL_I_ID4       INTEGER,
                IN   #OL_I_ID5       INTEGER,
                IN   #OL_I_ID6       INTEGER,
                IN   #OL_I_ID7       INTEGER,
                IN   #OL_I_ID8       INTEGER,
                IN   #OL_I_ID9       INTEGER,
                IN   #OL_I_ID10      INTEGER,
                IN   #OL_I_ID11      INTEGER,
                IN   #OL_I_ID12      INTEGER,
                IN   #OL_I_ID13      INTEGER,
                IN   #OL_I_ID14      INTEGER,
                IN   #OL_I_ID15      INTEGER,
                IN   #OL_I_ID_JOIN    VARCHAR(105) ,
                IN   #OL_QUANTITY_JOIN VARCHAR(60)  ,
                INOUT #S_JOIN         VARCHAR(1215),
                INOUT #I_JOIN         VARCHAR(1215),
                IN   #OL_SUPPLY_W_JOIN VARCHAR(60)
                )
NEWORDER:BEGIN
-- DECLARE
    DECLARE SQLSTATE        CHAR(5)      DEFAULT '00000';
    DECLARE SAPSTOP         CHAR(1)      DEFAULT '/';
    DECLARE @OL_I_ID        INTEGER;
    DECLARE @OL_SUPPLY_W_ID  SMALLINT;
    DECLARE @OL_QUANTITY     SMALLINT;
    DECLARE @S_QUANTITY      SMALLINT;
    DECLARE @I_PRICEH        SMALLINT;
    DECLARE @I_NAMEH         CHAR(24);
    DECLARE @I_DATAH         CHAR(50);
    DECLARE @S_DATA         CHAR(50);
    DECLARE @S_YTD          INTEGER;
    DECLARE @S_ORDER_CNT     SMALLINT;
    DECLARE @S_REMOTE_CNT     SMALLINT;
    DECLARE @D_NEXT_O_ID     INTEGER;
    DECLARE @OL_NUMBER       SMALLINT;
    DECLARE @STOCK_NUM       SMALLINT;
    DECLARE @MATCH_TBL_CNT    SMALLINT;
    DECLARE @S_DIST         CHAR(24);
    DECLARE @S_DIST_01       CHAR(24);
    DECLARE @S_DIST_02       CHAR(24);
    DECLARE @S_DIST_03       CHAR(24);
    DECLARE @S_DIST_04       CHAR(24);
    DECLARE @S_DIST_05       CHAR(24);
    DECLARE @S_DIST_06       CHAR(24);
    DECLARE @S_DIST_07       CHAR(24);
    DECLARE @S_DIST_08       CHAR(24);
    DECLARE @S_DIST_09       CHAR(24);
    DECLARE @S_DIST_10       CHAR(24);
    DECLARE @S_DIST_JOIN     CHAR(240)  ;
    DECLARE @C_OL_I_ID       CHAR(7)    ;
    DECLARE @C_I_PRICEH      CHAR(6)    ;
    DECLARE @C_S_QUANTITY     CHAR(6)    ;
    DECLARE @OL_AMOUNT       INTEGER    ;
    DECLARE @O_OL_CNT        SMALLINT   ;
    DECLARE @DIST_POS        SMALLINT   ;
    DECLARE @FILL_CNT        SMALLINT;
    DECLARE @TMP_CNT         SMALLINT;

-- (7) ITEM    table sele(IN)
    DECLARE ITEM_H CURSOR FOR
        SELECT I_PRICE,
            I_NAME,
            I_DATA,
            I_ID
         FROM  TPCC_SCHEMA.ITEM
```

```
                WHERE  TPCC_SCHEMA.ITEM.I_ID
                    IN( #OL_I_ID1    ,
                        #OL_I_ID2    ,
                        #OL_I_ID3    ,
                        #OL_I_ID4    ,
                        #OL_I_ID5    ,
                        #OL_I_ID6    ,
                        #OL_I_ID7    ,
                        #OL_I_ID8    ,
                        #OL_I_ID9    ,
                        #OL_I_ID10   ,
                        #OL_I_ID11   ,
                        #OL_I_ID12   ,
                        #OL_I_ID13   ,
                        #OL_I_ID14   ,
                        #OL_I_ID15   ) ;

-- (8) STOCK     table select
       DECLARE CNSS_HOME CURSOR FOR
          SELECT S_I_ID,S_QUANTITY,
             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_YTD,S_ORDER_CNT,S_REMOTE_CNT,S_DATA
           FROM  TPCC_SCHEMA.STOCK
           WHERE S_W_ID = #W_ID
            AND S_I_ID  IN( #OL_I_ID1    ,
                        #OL_I_ID2    ,
                        #OL_I_ID3    ,
                        #OL_I_ID4    ,
                        #OL_I_ID5    ,
                        #OL_I_ID6    ,
                        #OL_I_ID7    ,
                        #OL_I_ID8    ,
                        #OL_I_ID9    ,
                        #OL_I_ID10   ,
                        #OL_I_ID11   ,
                        #OL_I_ID12   ,
                        #OL_I_ID13   ,
                        #OL_I_ID14   ,
                        #OL_I_ID15   )
           ORDER BY S_I_ID DESC
             FOR UPDATE ;

   SET @DIST_POS    = 1+((#D_ID-1)*24);
   SET @O_OL_CNT    = #H_CNT + #R_CNT ;
   SET #O_ALL_LOCAL = 1             ;

   -- (4) CUSTOMER  table select
                WHENEVER SQLERROR  GOTO ERR_S_CM;
                    WHENEVER NOT FOUND GOTO
ERR_S_CM;
       SELECT C_LAST,C_CREDIT,C_DISCOUNT
          INTO #C_LAST,
             #C_CREDIT,
             #C_DISCOUNT
          FROM  TPCC_SCHEMA.CUSTOMER
          WHERE C_W_ID = #W_ID
           AND C_D_ID = #D_ID
           AND C_ID   = #C_ID;
                WHENEVER SQLERROR  CONTINUE;
                    WHENEVER NOT FOUND CONTINUE;
     IF  #H_CNT = 0 THEN
      GOTO REMORT_PROC ;
```

```
           END IF;

HOME_PROC:
-- Home Warehouse PROCESS START
-- (              Warehouse id              )
    -- (7) ITEM     table select
                        WHENEVER SQLERROR  GOTO ERR_S_IT;
                        WHENEVER NOT FOUND GOTO ERR_S_IT;
       OPEN  ITEM_H ;
                        WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;
-- LOOP
    SET  @MATCH_TBL_CNT = 0 ;
    INCNT:LOOP
                        WHENEVER SQLERROR  GOTO ERR_S_IT;
                        WHENEVER NOT FOUND GOTO L1;
        FETCH  ITEM_H
           INTO @I_PRICEH,
               @I_NAMEH,
               @I_DATAH,
               @OL_I_ID;
                        WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;

       SET  @MATCH_TBL_CNT = @MATCH_TBL_CNT + 1;

       SET  @C_I_PRICEH   = CAST(@I_PRICEH AS CHAR(6) )       ;
       SET  #I_JOIN       = #I_JOIN    || @C_I_PRICEH ||
                       @I_NAMEH   || @I_DATAH    || SAPSTOP ;

    END LOOP INCNT;
-- LOOP END

L1:  IF  @MATCH_TBL_CNT < #H_CNT  THEN
        SET   #ITEM_NF_CTR = -2 ;
        GOTO   NORMAL_END ;
     END IF;

     CLOSE  ITEM_H ;

--     (8) STOCK     table select
--     (9) STOCK     table update
                        WHENEVER SQLERROR  GOTO ERR_S_ST;
                        WHENEVER NOT FOUND GOTO ERR_S_ST;
       OPEN   CNSS_HOME  ;
                        WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;
-- LOOP
    SET   @STOCK_NUM = 0;
    OLCNT:LOOP
      IF @STOCK_NUM = #H_CNT THEN
         GOTO L3 ;
      END IF;

        -- (8) STOCK     table select
                        WHENEVER SQLERROR  GOTO ERR_S_ST;
                        WHENEVER NOT FOUND GOTO L3      ;
       FETCH  CNSS_HOME
          INTO @OL_I_ID,@S_QUANTITY,
             @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_YTD,@S_ORDER_CNT,@S_REMOTE_CNT,@S_DATA;
                        WHENEVER SQLERROR  CONTINUE;
```

```
                    WHENEVER NOT FOUND CONTINUE;
        SET  @S_DIST_JOIN = @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                        ;
        SET @S_DIST = SUBSTRING(@S_DIST_JOIN FROM
@DIST_POS FOR 24)    ;

        SET @OL_QUANTITY  =
CAST(SUBSTRING(#OL_QUANTITY_JOIN
                  FROM 1+(@STOCK_NUM * 4) FOR 4)
                  AS SMALLINT )              ;

        SET @S_QUANTITY = (@S_QUANTITY - @OL_QUANTITY);
        IF  @S_QUANTITY < 10 THEN
          SET @S_QUANTITY = @S_QUANTITY + 91  ;
        END IF;

        SET @S_YTD      = @S_YTD + @OL_QUANTITY;
        SET @S_ORDER_CNT = @S_ORDER_CNT + 1;

        -- (9) STOCK     table update
                    WHENEVER SQLERROR  GOTO ERR_U_ST;
        UPDATE TPCC_SCHEMA.STOCK
            SET S_QUANTITY  = @S_QUANTITY,
                S_YTD       = @S_YTD,
                S_ORDER_CNT  = @S_ORDER_CNT,
                S_REMOTE_CNT = @S_REMOTE_CNT
            WHERE CURRENT OF CNSS_HOME ;
                    WHENEVER SQLERROR  CONTINUE;

        SET  @C_S_QUANTITY  = CAST(@S_QUANTITY AS
CHAR(6) )        ;
        SET  #S_JOIN      = #S_JOIN   || @C_S_QUANTITY ||
                  @S_DIST   || @S_DATA    || SAPSTOP ;

        SET @STOCK_NUM = @STOCK_NUM + 1;

    END LOOP OLCNT;
-- LOOP END

L3:  IF    @STOCK_NUM <> #H_CNT
      AND @STOCK_NUM <> #ITEM_NF_CTR THEN
        GOTO ERR_S_ST_NF;
    END IF;
    CLOSE  CNSS_HOME  ;

--
-- LOOP
    SET  @FILL_CNT = 0 ;
    SET   @TMP_CNT = #H_CNT - @STOCK_NUM ;
    FILLCNT:LOOP
      IF  @FILL_CNT = @TMP_CNT THEN
          GOTO L4 ;
      END IF;

      SET  #I_JOIN      = #I_JOIN ||
```

```
                     '0    '||
                     '12345678901234567890 1234' ||
                     '123456789012345678901234567890123456789012 34567890' ||
                     SAPSTOP ;
        SET   #S_JOIN      = #S_JOIN ||
                     '123456' ||
                     '12345678901234567890 1234' ||
                     '123456789012345678901234567890123456789012 34567890' ||
                     SAPSTOP ;

        SET @FILL_CNT = @FILL_CNT + 1;

    END LOOP FILLCNT;
-- LOOP END

-- Home Warehouse PROCESS END

L4:  IF  #R_CNT = 0 THEN
        GOTO DISTRICT_PROC ;
    END IF;

REMORT_PROC:
-- Remote Warehouse process start
-- (                       Warehouse)
-- LOOP
    SET  @MATCH_TBL_CNT = 0 ;
    SET   @STOCK_NUM     = 0 ;

    OLCNT_R:LOOP
R1:    IF @STOCK_NUM = #R_CNT THEN
        GOTO R3 ;
      END IF;

      SET  @OL_I_ID       = CAST(SUBSTRING(#OL_I_ID_JOIN
                  FROM 1+(@STOCK_NUM * 7) FOR 7)
                  AS INTEGER )                  ;

      SET  @OL_SUPPLY_W_ID = CAST(SUBSTRING(#OL_SUPPLY_W_JOIN
                  FROM 1+(@STOCK_NUM * 4) FOR 4)
                  AS SMALLINT )                 ;
      -- (7) ITEM     table select
                    WHENEVER SQLERROR  GOTO ERR_S_IT ;
                    WHENEVER NOT FOUND GOTO R4       ;
      SELECT I_PRICE,I_NAME,I_DATA
          INTO  @I_PRICEH,
               @I_NAMEH ,
               @I_DATAH
          FROM   TPCC_SCHEMA.ITEM
          WHERE  I_ID = @OL_I_ID ;
                    WHENEVER SQLERROR  CONTINUE;
                    WHENEVER NOT FOUND CONTINUE;

      SET   @MATCH_TBL_CNT = @MATCH_TBL_CNT + 1    ;

      SET  @C_I_PRICEH    = CAST(@I_PRICEH AS CHAR(6) )      ;
      SET  #I_JOIN      = #I_JOIN   || @C_I_PRICEH ||
                  @I_NAMEH  || @I_DATAH    || SAPSTOP ;

      -- (8) STOCK     table select
                    WHENEVER SQLERROR  GOTO ERR_S_ST;
                    WHENEVER NOT FOUND GOTO ERR_S_ST;
      SELECT  S_QUANTITY,
          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,         -- LOOP END
            S_YTD,S_ORDER_CNT,S_REMOTE_CNT,S_DATA                       R3:
        INTO @S_QUANTITY,                                                   IF  @MATCH_TBL_CNT < #R_CNT  THEN
                                                                             IF #ITEM_NF_CTR = -1 THEN
@S_DIST_01,@S_DIST_02,@S_DIST_03,@S_DIST_04,@S_DIST_05,                          SET   #ITEM_NF_CTR = @MATCH_TBL_CNT ;
                                                                             ELSE
@S_DIST_06,@S_DIST_07,@S_DIST_08,@S_DIST_09,@S_DIST_10,                          SET   #ITEM_NF_CTR = #ITEM_NF_CTR + @MATCH_TBL_CNT ;
        @S_YTD,@S_ORDER_CNT,@S_REMOTE_CNT,@S_DATA                           END IF;
        FROM  TPCC_SCHEMA.STOCK                                           END IF;
        WHERE S_W_ID = @OL_SUPPLY_W_ID
         AND S_I_ID = @OL_I_ID  ;                                      -- Remote Warehouse process end
                        WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;                   DISTRICT_PROC:
                                                                          -- (3) DISTRICT  table update
        SET  @S_DIST_JOIN = @S_DIST_01                                                    WHENEVER SQLERROR  GOTO ERR_U_DI;
                    || @S_DIST_02                                                         WHENEVER NOT FOUND GOTO ERR_U_DI;
                    || @S_DIST_03                                          UPDATE TPCC_SCHEMA.DISTRICT
                    || @S_DIST_04                                             SET   D_NEXT_O_ID = D_NEXT_O_ID+1
                    || @S_DIST_05                                             WHERE D_W_ID     = #W_ID
                    || @S_DIST_06                                              AND D_ID       = #D_ID ;
                    || @S_DIST_07                                                         WHENEVER SQLERROR  CONTINUE;
                    || @S_DIST_08                                                         WHENEVER NOT FOUND CONTINUE;
                    || @S_DIST_09                                          -- (2) DISTRICT  table select
                    || @S_DIST_10                      ;
        SET  @S_DIST = SUBSTRING(@S_DIST_JOIN FROM                         SELECT  D_NEXT_O_ID-1,D_TAX
@DIST_POS FOR 24)   ;                                                          INTO  #O_ID ,#D_TAX
                                                                              FROM  TPCC_SCHEMA.DISTRICT
        SET  @OL_QUANTITY  =                                                   WHERE D_W_ID = #W_ID
CAST(SUBSTRING(#OL_QUANTITY_JOIN                                                AND D_ID  = #D_ID ;
                    FROM 1+((@STOCK_NUM+#H_CNT) * 4) FOR 4)
                    AS SMALLINT )                    ;                     -- (6) ORDERS    table insert
                                                                                         WHENEVER SQLERROR  GOTO ERR_I_OR;
        SET @S_QUANTITY = (@S_QUANTITY - @OL_QUANTITY);                                   WHENEVER NOT FOUND GOTO ERR_I_OR;
        IF  @S_QUANTITY < 10 THEN                                          INSERT INTO TPCC_SCHEMA.ORDERS
            SET @S_QUANTITY = @S_QUANTITY + 91  ;                              VALUES (#O_ID,
        END IF;                                                                   #D_ID,
                                                                                  #W_ID,
        SET @S_YTD      = @S_YTD + @OL_QUANTITY;                                   #C_ID,
        SET @S_ORDER_CNT = @S_ORDER_CNT + 1;                                      #O_ENTRY_D,
        IF  @OL_SUPPLY_W_ID <> #W_ID THEN                                          NULL,
            SET @S_REMOTE_CNT = @S_REMOTE_CNT + 1;                                 @O_OL_CNT,
            SET #O_ALL_LOCAL  = 0;                                                 #O_ALL_LOCAL);
        END IF;                                                                               WHENEVER SQLERROR  CONTINUE;
                                                                                              WHENEVER NOT FOUND CONTINUE;
       -- (9) STOCK     table update                                      -- (5) NEWORDER  table insert
                   WHENEVER SQLERROR  GOTO ERR_U_ST;                                      WHENEVER SQLERROR  GOTO ERR_I_NO;
        UPDATE TPCC_SCHEMA.STOCK                                                          WHENEVER NOT FOUND GOTO ERR_I_NO;
        SET  S_QUANTITY   = @S_QUANTITY,                                   INSERT INTO TPCC_SCHEMA.NEWORDER
            S_YTD        = @S_YTD,                                             VALUES (#O_ID,
            S_ORDER_CNT  = @S_ORDER_CNT,                                           #D_ID,
            S_REMOTE_CNT = @S_REMOTE_CNT                                           #W_ID);
         WHERE   S_W_ID = @OL_SUPPLY_W_ID                                                     WHENEVER SQLERROR  CONTINUE;
          AND   S_I_ID = @OL_I_ID  ;                                                          WHENEVER NOT FOUND CONTINUE;
                        WHENEVER SQLERROR  CONTINUE;                       -- (1) WAREHOUSE table update
                                                                                         WHENEVER SQLERROR  GOTO ERR_S_WH;
        SET  @C_S_QUANTITY  = CAST(@S_QUANTITY AS CHAR(6)                  SELECT  W_TAX
)   ;                                                                          INTO  #W_TAX
        SET  #S_JOIN      = #S_JOIN   || @C_S_QUANTITY ||                      FROM  TPCC_SCHEMA.WAREHOUSE
                    @S_DIST   || @S_DATA     || SAPSTOP ;                      WHERE W_ID=#W_ID ;
                                                                                         WHENEVER SQLERROR  CONTINUE;
R2:    SET @STOCK_NUM = @STOCK_NUM + 1 ;

    END LOOP OLCNT_R;                                                      NORMAL_END:
```

```
        SET #STATE  = '00000' ;                                    SET #STATE  = SQLSTATE;
        LEAVE NEWORDER ;
                                                            END NEWORDER
--                                                          END-EXEC;
--                                                          :::::::::::::::
R4:     SET  #I_JOIN      = #I_JOIN ||                       Y_ODERSTAT
                    '0      '||                              :::::::::::::::
                    '12345678901234567890... ' ||            -- /*********STORED PROCEDURE*****************************************/
                                                            -- /** Y_ODERSTAT   COPYRIGHT FUJITSU LIMITED 1997           **/
'1234567890123456789012345678901234567890' ||               -- /**    :                              **/
                    SAPSTOP ;                               -- /**    :                              **/
        SET  #S_JOIN      = #S_JOIN ||                       -- /**    :SymfoWARE RDB TPC-C Benchmark          **/
                    '123456' ||                             -- /**    :  Order-Status                **/
                    '12345678901234567890' ||               -- /**    :1996/10/12                   **/
                                                            -- /**     1997/03/13  Revision3.3 : Any Error(Clause 2.3.6)    **/
'1234567890123456789012345678901234567890' ||               -- /****************************************************************/
                    SAPSTOP ;
        GOTO R2 ;                                           -- #OL_JOIN         VARCHAR(570)
                                                            -- +-----------------------------------+
                                                            -- | sqllen        short          |
--SQLERR:NOT_FOUND:                                          -- +-sqlver----------------------------+ ---
ERR_I_OR:                                                    -- | #OL_I_IDn      CHAR(7)        |  |
        SET #ERRPOS = 107    ;                               -- +-----------------------------------+  |
        SET #STATE  = SQLSTATE;                              -- | #OL_AMOUNTn     CHAR(8)       |  |
        LEAVE NEWORDER       ;                               -- +-----------------------------------+  |
ERR_I_OL:                                                    -- | #OL_SUPPLY_W_IDn CHAR(4)      |  |
        SET #ERRPOS = 108    ;                               -- +-----------------------------------+  |
        SET #STATE  = SQLSTATE;                              -- | #OL_QUANTITYn   CHAR(4)       |  |
        LEAVE NEWORDER       ;                               -- +-----------------------------------+  |
ERR_I_NO:                                                    -- | #OL_DELIVERYn   CHAR(14)      |  |
        SET #ERRPOS = 109    ;                               -- +-----------------------------------+  |
        SET #STATE  = SQLSTATE;                              -- |            CHAR(1) "/"       |  |
        LEAVE NEWORDER       ;                               -- +-----------------------------------+ ---
ERR_S_IT:                                                    --
        SET #ERRPOS = 201    ;                               -- +-----------------------------------+
        SET #STATE  = SQLSTATE;                              --
        LEAVE NEWORDER       ;                               --
ERR_S_WH:                                                    -- ***********************************************************************
        SET #ERRPOS = 202    ;                               EXEC SQL
        SET #STATE  = SQLSTATE;                              CREATE PROCEDURE TPCC_SCHEMA.Y_ODERSTAT(OUT #STATE
        LEAVE NEWORDER       ;                               CHAR(5) ,
ERR_S_DI:                                                               INOUT #ERRPOS        INTEGER ,
        SET #ERRPOS = 203    ;                                          IN   #W_ID        SMALLINT,
        SET #STATE  = SQLSTATE;                                         IN   #D_ID        SMALLINT,
        LEAVE NEWORDER       ;                                          INOUT #C_ID          INTEGER ,
ERR_S_ST:                                                               OUT  #C_FIRST       CHAR(16),
        SET #ERRPOS = 204    ;                                          OUT  #C_MIDDLE      CHAR(2) ,
        SET #STATE  = SQLSTATE;                                         INOUT #C_LAST        CHAR(16),
        LEAVE NEWORDER       ;                                          OUT  #C_BALANCE      DOUBLE PRECISION,
ERR_S_ST_NF:                                                            INOUT #O_ID          INTEGER ,
        SET #ERRPOS = 204    ;                                          OUT  #O_ENTRY_D      CHAR(14),
        SET #STATE  = '02000' ;                                         OUT  #O_CARRIER_ID    SMALLINT,
        LEAVE NEWORDER       ;                                          INOUT #O_OL_CNT       SMALLINT,
ERR_S_CM:                                                               INOUT #OL_JOIN        VARCHAR(570)
        SET #ERRPOS = 205    ;                                          )
        SET #STATE  = SQLSTATE;                              ORDER_STATUS:BEGIN
        LEAVE NEWORDER       ;                               -- DECLARE
ERR_U_DI:                                                        DECLARE SQLSTATE         CHAR(5)      DEFAULT '00000';
        SET #ERRPOS = 303    ;                                   DECLARE SAPSTOP          CHAR(1)      DEFAULT '/';
        SET #STATE  = SQLSTATE;                                  DECLARE DELIVERY_D       CHAR(14)     DEFAULT '777777777';
        LEAVE NEWORDER       ;                                   DECLARE @OL_I_ID         INTEGER;
ERR_U_ST:                                                        DECLARE @OL_SUPPLY_W_ID   SMALLINT;
        SET #ERRPOS = 304    ;                                   DECLARE @OL_QUANTITY     SMALLINT;
```

```
    DECLARE @OL_AMOUNT      INTEGER;
    DECLARE @OL_DELIVERY_D   CHAR(14);
    DECLARE @OL_NUMBER       INTEGER;
    DECLARE @NAMECOUNT       INTEGER;
    DECLARE @J              INTEGER;
    DECLARE @I              INTEGER;
    DECLARE @WORK           VARCHAR(100);

-- DEFINE  CUSTOMER  table cursor
    DECLARE COCS  CURSOR FOR
        SELECT C_ID,
            C_FIRST,
            C_MIDDLE,
            C_LAST,
            C_BALANCE
            FROM  TPCC_SCHEMA.CUSTOMER
            WHERE C_LAST = #C_LAST
              AND C_W_ID = #W_ID
              AND C_D_ID = #D_ID
            ORDER BY C_FIRST;

-- DEFINE  ORDERLINE table cursor
    DECLARE COOLS CURSOR FOR
        SELECT OL_I_ID,
            OL_SUPPLY_W_ID,
            OL_DELIVERY_D,
            OL_QUANTITY,
            OL_AMOUNT
            FROM  TPCC_SCHEMA.ORDERLINE
            WHERE OL_W_ID = #W_ID
              AND OL_D_ID = #D_ID
              AND OL_O_ID = #O_ID
              AND OL_NUMBER
IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);
--$
--$            WHERE OL_W_ID = #W_ID
--$              AND OL_D_ID = #D_ID
--$              AND OL_O_ID = #O_ID;

    IF #C_ID = 0 THEN
        -- Customer Last Name Payment Transaction
        -- (1) CUSTOMER  table select
                WHENEVER SQLERROR  GOTO ERR_S_CM;
                WHENEVER NOT FOUND GOTO
ERR_S_CM;
        SELECT COUNT(*)
            INTO  @NAMECOUNT
        FROM  TPCC_SCHEMA.CUSTOMER
        WHERE C_LAST = #C_LAST
          AND C_W_ID = #W_ID
          AND C_D_ID = #D_ID;
                WHENEVER SQLERROR  CONTINUE;
                WHENEVER NOT FOUND CONTINUE;
        IF @NAMECOUNT > 0 THEN
                WHENEVER SQLERROR  GOTO ERR_S_CM;
                WHENEVER NOT FOUND GOTO
ERR_S_CM;
        OPEN  COCS;
                WHENEVER SQLERROR  CONTINUE;
                WHENEVER NOT FOUND CONTINUE;
        SET   @J = @NAMECOUNT + 1;
        SET   @J = @J      / 2;
        SET   @I = 0          ;
```

```
    NAMECNT:LOOP
      IF  @I = @J THEN
        LEAVE NAMECNT ;
      END IF;
      SET   @I = @I + 1  ;
                WHENEVER SQLERROR  GOTO ERR_S_CM;
                WHENEVER NOT FOUND GOTO ERR_S_CM;
        FETCH  COCS
            INTO #C_ID,
                #C_FIRST,
                #C_MIDDLE,
                #C_LAST,
                #C_BALANCE;
                WHENEVER SQLERROR  CONTINUE;
                WHENEVER NOT FOUND CONTINUE;
    END LOOP NAMECNT;
    CLOSE COCS;
  ELSE
    GOTO ERR_S_CM_NAME ;
  END IF;

ELSE
    -- Customer id Payment Transaction
    -- (2) CUSTOMER  table select
                WHENEVER SQLERROR  GOTO ERR_S_CM;
                WHENEVER NOT FOUND GOTO ERR_S_CM;
    SELECT C_FIRST,C_MIDDLE,C_LAST,C_BALANCE
        INTO  #C_FIRST,
            #C_MIDDLE,
            #C_LAST,
            #C_BALANCE
        FROM  TPCC_SCHEMA.CUSTOMER
        WHERE C_ID   = #C_ID
          AND C_D_ID = #D_ID
          AND C_W_ID = #W_ID;
                WHENEVER SQLERROR  CONTINUE;
                WHENEVER NOT FOUND CONTINUE;
END IF;

-- (3) ORDER    table select get max o_id record
                WHENEVER SQLERROR  GOTO ERR_S_OR;
                WHENEVER NOT FOUND GOTO ERR_S_OR;
SELECT   O_ID,
      O_ENTRY_D,
      O_CARRIER_ID,
      O_OL_CNT
      INTO  #O_ID,
          #O_ENTRY_D,
          #O_CARRIER_ID,
          #O_OL_CNT
      FROM  TPCC_SCHEMA.ORDERS
      WHERE O_ID   = (SELECT MAX(O_ID)
                FROM  TPCC_SCHEMA.ORDERS
                WHERE O_W_ID = #W_ID
                  AND O_D_ID = #D_ID
                  AND O_C_ID = #C_ID )
        AND O_W_ID = #W_ID
        AND O_D_ID = #D_ID
        AND O_C_ID = #C_ID;
                WHENEVER SQLERROR  GOTO ERR_S_OL;
                WHENEVER NOT FOUND GOTO ERR_S_OL;
OPEN    COOLS ;
                WHENEVER SQLERROR  CONTINUE;
```

```
                        WHENEVER NOT FOUND CONTINUE;
-- LOOP
    SET   @OL_NUMBER = 1;
    OLCNT:LOOP
     IF  #O_OL_CNT < @OL_NUMBER THEN
        LEAVE OLCNT ;
     END IF;

    -- (4) ORDER-LINE table select
                    WHENEVER SQLERROR  GOTO ERR_S_OL;
                    WHENEVER NOT FOUND GOTO
ERR_S_OL;
        FETCH  COOLS
            INTO @OL_I_ID,
                @OL_SUPPLY_W_ID,
                @OL_DELIVERY_D,
                @OL_QUANTITY,
                @OL_AMOUNT;
                    WHENEVER SQLERROR  CONTINUE;
                    WHENEVER NOT FOUND CONTINUE;
        IF @OL_DELIVERY_D IS NULL THEN
           SET @WORK   = CAST(@OL_I_ID       AS CHAR(7))
                || CAST(@OL_AMOUNT      AS CHAR(8))
                || CAST(@OL_SUPPLY_W_ID AS CHAR(4))
                || CAST(@OL_QUANTITY    AS CHAR(4))
                ||        DELIVERY_D  || SAPSTOP   ;
        ELSE
           SET @WORK   = CAST(@OL_I_ID       AS CHAR(7))
                || CAST(@OL_AMOUNT      AS CHAR(8))
                || CAST(@OL_SUPPLY_W_ID AS CHAR(4))
                || CAST(@OL_QUANTITY    AS CHAR(4))
                ||      @OL_DELIVERY_D  || SAPSTOP   ;
        END IF ;

        SET #OL_JOIN = #OL_JOIN || @WORK;

        SET @OL_NUMBER  = @OL_NUMBER + 1;

    END LOOP OLCNT;
-- LOOP END

    CLOSE COOLS       ;

    COMMIT WORK      ;
    SET #STATE = '00000' ;
    LEAVE ORDER_STATUS  ;

--SQLERR:NOT_FOUND:
ERR_S_CM_NAME:
    SET #ERRPOS = 205   ;
    SET #STATE = '02000' ;
    ROLLBACK WORK      ;
    LEAVE ORDER_STATUS   ;
ERR_S_CM:
    SET #ERRPOS = 205    ;
    SET #STATE = SQLSTATE;
    ROLLBACK WORK      ;
    LEAVE ORDER_STATUS   ;
ERR_S_OR:
    SET #ERRPOS = 207    ;
    SET #STATE = SQLSTATE;
    ROLLBACK WORK      ;
    LEAVE ORDER_STATUS   ;
```

```
ERR_S_OL:
    SET #ERRPOS = 208    ;
    SET #STATE = SQLSTATE;
    ROLLBACK WORK       ;

END ORDER_STATUS
END-EXEC;
:::::::::::::
Y_PAYMENT_cent.h10.out.4
:::::::::::::
-- /**********STORED PROCEDURE*****************************************/
-- /** Y_PAYMENT    COPYRIGHT FUJITSU LIMITED 1997         **/
-- /**    :                                     **/
-- /**    :                                     **/
-- /**    :SymfoWARE RDB TPC-C Benchmark              **/
-- /**    :   Payment                         **/
-- /**    :1996/10/12                          **/
-- /**    1997/03/13  Revision3.3 : Any Error(Clause 2.3.6)     **/
-- /*****************************************************************/

EXEC SQL
CREATE PROCEDURE TPCC_SCHEMA.Y_PAYMENT_H10_OUT4(OUT #STATE
CHAR(5) ,
                        INOUT #ERRPOS        INTEGER ,
                        IN   #W_ID          SMALLINT,
                        IN   #D_ID          SMALLINT,
                        INOUT #C_ID          INTEGER ,
                        IN   #C_D_ID         SMALLINT,
                        IN   #C_W_ID         SMALLINT,
                        IN   #H_AMOUNT        INTEGER ,
                        IN   #H_DATE         CHAR(14),
                        INOUT #W_NAME        CHAR(10),
                        OUT  #W_STREET_1     CHAR(20),
                        OUT  #W_STREET_2     CHAR(20),
                        OUT  #W_CITY         CHAR(20),
                        OUT  #W_STATE        CHAR(2),
                        OUT  #W_ZIP         CHAR(9),
                        INOUT #D_NAME        CHAR(10),
                        OUT  #D_STREET_1     CHAR(20),
                        OUT  #D_STREET_2     CHAR(20),
                        OUT  #D_CITY         CHAR(20),
                        OUT  #D_STATE        CHAR(2),
                        OUT  #D_ZIP         CHAR(9),
                        OUT  #C_FIRST        CHAR(16),
                        OUT  #C_MIDDLE       CHAR(2),
                        INOUT #C_LAST        CHAR(16),
                        OUT  #C_STREET_1     CHAR(20),
                        OUT  #C_STREET_2     CHAR(20),
                        OUT  #C_CITY         CHAR(20),
                        OUT  #C_STATE        CHAR(2),
                        OUT  #C_ZIP         CHAR(9),
                        OUT  #C_PHONE        CHAR(16),
                        INOUT #C_CREDIT       CHAR(2),
                        OUT  #C_CREDIT_LIM    DECIMAL(12,0), --98.11.06
                        OUT  #C_DISCOUNT      SMALLINT,
                        OUT #C_BALANCE        DECIMAL(12,0), --98.11.06
                        OUT #C_YTD_PAYMENT   DECIMAL(12,0), --98.11.06
                        INOUT #C_PAYMENT_CNT   SMALLINT,
                        OUT  #C_SINCE        CHAR(14),
                        INOUT #C_DATA        VARCHAR(500)
                        )

PAYMENT:BEGIN
```

```
-- DECLARE
        DECLARE @C_BALANCE        DECIMAL(12,0); --98.11.06
+oza
      DECLARE @C_YTD_PAYMENT   DECIMAL(12,0); --98.11.06
+oza
--          DECLARE @C_DATA        VARCHAR(500);  --98.11.06
+oza
      DECLARE SQLSTATE         CHAR(5)         DEFAULT '00000';
      DECLARE @CNT             INTEGER;
      DECLARE @NAMECOUNT       INTEGER;
      DECLARE @W_YTD           DECIMAL(12,0);  --98.11.06
      DECLARE @D_YTD           DECIMAL(12,0);  --98.11.06
      DECLARE @C_DATA474       CHAR(474);      --98.10.13 change
      DECLARE @H_DATA          CHAR(24);
      DECLARE @H_AMOUNT        DECIMAL(10,0);   --98.11.06
+oza

-- CUSTOMER
      DECLARE CPCS CURSOR FOR
            SELECT C_ID,
                   C_FIRST,
                   C_MIDDLE,
                   C_LAST,
                   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_YTD_PAYMENT,
                   C_PAYMENT_CNT
            FROM  TPCC_SCHEMA.CUSTOMER
            WHERE C_LAST = #C_LAST
              AND C_W_ID = #C_W_ID
              AND C_D_ID = #C_D_ID
            ORDER BY C_FIRST;

      IF #C_ID = 0 THEN
        -- Customer Last Name process
        -- (5) CUSTOMER  table select
                    WHENEVER SQLERROR  GOTO ERR_S_CM;
                        WHENEVER NOT FOUND GOTO
ERR_S_CM;
          SELECT COUNT(*) INTO @NAMECOUNT
              FROM TPCC_SCHEMA.CUSTOMER
              WHERE C_LAST = #C_LAST
              AND C_W_ID = #C_W_ID
              AND C_D_ID = #C_D_ID;
                    WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;
        -- (6) CUSTOMER
        -- Customer Last Name            C_FIRST
        -- NAMECOUNT/
        IF @NAMECOUNT > 0 THEN
          SET  @CNT      = @NAMECOUNT + 1;
          SET  @CNT      = @CNT       / 2;
          SET  @NAMECOUNT = @CNT          ;
                    WHENEVER SQLERROR  GOTO ERR_S_CM;

                    WHENEVER NOT FOUND GOTO ERR_S_CM;
      OPEN  CPCS;
                    WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;
      SET  @CNT  = 0 ;
      WHILE  @CNT < @NAMECOUNT DO
        SET  @CNT = @CNT + 1;
          -- (6) CUSTOMER  table
                    WHENEVER SQLERROR  GOTO ERR_S_CM;
                        WHENEVER NOT FOUND GOTO ERR_S_CM;
          FETCH CPCS
              INTO #C_ID,
                   #C_FIRST,
                   #C_MIDDLE,
                   #C_LAST,
                   #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_YTD_PAYMENT,
                   #C_PAYMENT_CNT;
                    WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;
        END WHILE;

        CLOSE CPCS;

      ELSE
        GOTO ERR_S_CM_NAME;
      END IF;
    ELSE
      -- C-ID PROCESS
      -- (7) CUSTOMER  teble
                    WHENEVER SQLERROR  GOTO ERR_S_CM;
                        WHENEVER NOT FOUND GOTO ERR_S_CM;
      SELECT C_FIRST,
             C_MIDDLE,
             C_LAST,
             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_YTD_PAYMENT,
             C_PAYMENT_CNT
      INTO #C_FIRST,
           #C_MIDDLE,
           #C_LAST,
           #C_STREET_1,
           #C_STREET_2,
```

```
        #C_CITY,                                              WHENEVER NOT FOUND GOTO ERR_U_CM;
        #C_STATE,                                 UPDATE   TPCC_SCHEMA.CUSTOMER
        #C_ZIP,                                      SET   C_BALANCE    = @C_BALANCE,
        #C_PHONE,                                          C_YTD_PAYMENT = @C_YTD_PAYMENT,
        #C_SINCE,                                          C_PAYMENT_CNT = #C_PAYMENT_CNT,
        #C_CREDIT,                                         C_DATA        = #C_DATA
        #C_CREDIT_LIM,                             WHERE C_ID        = #C_ID
        #C_DISCOUNT,                                 AND C_D_ID      = #C_D_ID
        @C_BALANCE,                                  AND C_W_ID      = #C_W_ID;
        @C_YTD_PAYMENT,                                     WHENEVER SQLERROR  CONTINUE;
        #C_PAYMENT_CNT                                      WHENEVER NOT FOUND CONTINUE;
     FROM  TPCC_SCHEMA.CUSTOMER               ELSE
     WHERE C_W_ID = #C_W_ID                   --*************--
       AND C_D_ID = #C_D_ID                   -- Good Customer
       AND C_ID  = #C_ID;                     --*************--
                  WHENEVER SQLERROR  CONTINUE; -- (8)GC-1 CUSTOMER table update
                  WHENEVER NOT FOUND CONTINUE;          WHENEVER SQLERROR  GOTO ERR_U_CM;
     END IF;                                            WHENEVER NOT FOUND GOTO ERR_U_CM;
                                              UPDATE TPCC_SCHEMA.CUSTOMER
                                                 SET   C_BALANCE    = @C_BALANCE,
--                                                      C_YTD_PAYMENT = @C_YTD_PAYMENT,
     SET @H_AMOUNT = #H_AMOUNT ;                        C_PAYMENT_CNT = #C_PAYMENT_CNT
                                                 WHERE C_ID   = #C_ID
     -- Customer          #C_BALANCE              AND C_D_ID = #C_D_ID
     -- Customer          #C_YTD_PAYMENT          AND C_W_ID = #C_W_ID;
     -- Customer          #C_PAYMENT_CNT                   WHENEVER SQLERROR  CONTINUE;
     SET @C_BALANCE    = @C_BALANCE    - @H_AMOUNT ;      WHENEVER NOT FOUND CONTINUE;
     SET @C_YTD_PAYMENT = @C_YTD_PAYMENT +      END IF;
@H_AMOUNT ;
     SET #C_PAYMENT_CNT = #C_PAYMENT_CNT + 1       ;   -- (3) DISTRICT table select
                                                        WHENEVER SQLERROR  GOTO ERR_S_DI;
     --(8) HISTORY teble insert                         WHENEVER NOT FOUND GOTO ERR_S_DI;
     IF #C_CREDIT = 'BC' THEN                  SELECT   D_NAME,
        --*************--                              D_STREET_1,
        -- Bad Customer                               D_STREET_2,
        --*************--                             D_CITY,
        -- (8)BC-1 CUSTOMER table select              D_STATE,
                  WHENEVER SQLERROR  GOTO             D_ZIP,
ERR_S_CM;                                             D_YTD
                  WHENEVER NOT FOUND GOTO        INTO  #D_NAME,
ERR_S_CM;                                             #D_STREET_1,
        SELECT C_DATA                                 #D_STREET_2,
           INTO  @C_DATA474                           #D_CITY,
           FROM  TPCC_SCHEMA.CUSTOMER                 #D_STATE,
           WHERE C_ID   = #C_ID                       #D_ZIP,
             AND C_D_ID = #C_D_ID                     @D_YTD
             AND C_W_ID = #C_W_ID;             FROM  TPCC_SCHEMA.DISTRICT
                  WHENEVER SQLERROR  CONTINUE;   WHERE D_ID   = #D_ID
                  WHENEVER NOT FOUND CONTINUE;     AND D_W_ID = #W_ID;
        -- (8)BC-2      c_data        .                 WHENEVER SQLERROR  CONTINUE;
        SET   #C_DATA   = CAST(#C_ID   AS CHAR(5))       WHENEVER NOT FOUND CONTINUE;
                  || CAST(#C_D_ID   AS CHAR(2))   -- (4) DISTRICT
                  || CAST(#C_W_ID   AS CHAR(4))   --98.10.13  SET   @D_YTD = @D_YTD + @H_AMOUNT;
change                                                  WHENEVER SQLERROR  GOTO ERR_U_DI;
                  || CAST(#D_ID     AS CHAR(2))         WHENEVER NOT FOUND GOTO ERR_U_DI;
                || CAST(#W_ID    AS CHAR(4))   --98.10.13 change UPDATE  TPCC_SCHEMA.DISTRICT
                  || CAST(#H_AMOUNT AS CHAR(7))     SET   D_YTD = @D_YTD
                  || ' '                            WHERE D_ID   = #D_ID
                  || @C_DATA474;                      AND D_W_ID = #W_ID;
                                                          WHENEVER SQLERROR  CONTINUE;
        -- (8) BC-3 CUSTOMER table update                WHENEVER NOT FOUND CONTINUE;
                  WHENEVER SQLERROR  GOTO         -- (1) WAREHOUSE
ERR_U_CM;                                                 WHENEVER SQLERROR  GOTO ERR_S_WH;
```

```
    SELECT  W_NAME,
            W_STREET_1,
            W_STREET_2,
            W_CITY,
            W_STATE,
            W_ZIP,
            W_YTD
        INTO #W_NAME,
            #W_STREET_1,
            #W_STREET_2,
            #W_CITY,
            #W_STATE,
            #W_ZIP,
            @W_YTD
        FROM  TPCC_SCHEMA.WAREHOUSE
        WHERE W_ID = #W_ID;
                        WHENEVER SQLERROR  CONTINUE;
    -- (2) WAREHOUSE
    SET    @W_YTD = @W_YTD + @H_AMOUNT;
                        WHENEVER SQLERROR  GOTO
ERR_U_WH;
    UPDATE  TPCC_SCHEMA.WAREHOUSE
            SET  W_YTD = @W_YTD
            WHERE W_ID  = #W_ID;
                        WHENEVER SQLERROR  CONTINUE;
    -- (9) HISTORY              #H_DATA
    --    HISTORY
    SET    @H_DATA = #W_NAME || ' ' || #D_NAME;
                        WHENEVER SQLERROR  GOTO ERR_I_HI;
                        WHENEVER NOT FOUND GOTO ERR_I_HI;
    INSERT
        INTO   TPCC_SCHEMA.HISTORY
            (H_C_ID,
             H_C_D_ID,
             H_C_W_ID,
             H_D_ID,
             H_W_ID,
             H_DATE,
             H_AMOUNT,
             H_DATA)
        VALUES (#C_ID,
             #C_D_ID,
             #C_W_ID,
             #D_ID,
             #W_ID,
             #H_DATE,
             #H_AMOUNT,
             @H_DATA);
                        WHENEVER SQLERROR  CONTINUE;
                        WHENEVER NOT FOUND CONTINUE;
    SET #C_BALANCE = @C_BALANCE;
    SET #C_YTD_PAYMENT = @C_YTD_PAYMENT;
--    SET #C_DATA = @C_DATA;
    COMMIT WORK        ;
    SET #STATE = '00000' ;
    LEAVE PAYMENT        ;

--SQLERR:NOT_FOUND:
ERR_I_HI:
    SET #ERRPOS = 106    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
```

```
ERR_S_WH:
    SET #ERRPOS = 202    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
ERR_S_DI:
    SET #ERRPOS = 203    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
ERR_S_CM_NAME:
    SET #ERRPOS = 205    ;
    SET #STATE  = '02000' ;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
ERR_S_CM:
    SET #ERRPOS = 205    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
ERR_U_WH:
    SET #ERRPOS = 302    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
ERR_U_DI:
    SET #ERRPOS = 303    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE PAYMENT        ;
ERR_U_CM:
    SET #ERRPOS = 305    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;

END PAYMENT
END-EXEC;
::::::::::::
Y_stored_cent.sh
::::::::::::
 set -x
rdbddlex -d TPCC  -x Y_NORDER
#rdbddlex -d TPCC  -x Y_NORDER4    #99.06.01          NG
rdbddlex -d TPCC  -x Y_NORDER5    #99.06.10          NG
rdbddlex -d TPCC  -x Y_PAYMENT_cent #
rdbddlex -d TPCC  -x Y_ODERSTAT
rdbddlex -d TPCC  -x Y_DELIVERY    #          (      )
rdbddlex -d TPCC  -x Y_STOCKLV

::::::::::::
Y_stored_cent.sh.h10.out4
::::::::::::
 set -x

rdbddlex -d TPCC  -x Y_PAYMENT_cent.h10.out.4 #        wk err detail
rdbrcp


::::::::::::
crta.def.cent
::::::::::::
```

≡ B

```
--
##########################################################
#############
-- #   TPC-C
-- #
-- #   1995.5.15
-- #   1996.4.18     DECIMAL -> CHAR or SMALLINT or INTEGER
-- #   1996.10.18  C_ID,H_C_ID,O_C_ID SMALLINT -> INTEGER
-- #                                   I_IM_ID
-- #   1998.11.24  W_YTD, D_YTD, C_BALANCE, C_CREDIT_LIM,
-- #                   C_YTD_PAYMENT      ->    12
-- #
--
##########################################################
#############
CREATE SCHEMA TPCC_SCHEMA
--
##########################################################
#############
-- #
--
##########################################################
#############
CREATE TABLE TPCC_SCHEMA.WAREHOUSE(
        W_ID        SMALLINT NOT NULL,
        W_NAME CHAR(10) NOT NULL,
        W_STREET_1 CHAR(20) NOT NULL,
        W_STREET_2 CHAR(20) NOT NULL,
        W_CITY   CHAR(20) NOT NULL,
        W_STATE CHAR(2)   NOT NULL,
        W_ZIP    CHAR(9)   NOT NULL,
--      W_TAX    DECIMAL(4,4) NOT NULL,'96/04/18
        W_TAX    SMALLINT NOT NULL,
--      W_YTD    DECIMAL(12,2) NOT NULL, '98/11/24
        W_YTD    DECIMAL(12,0) NOT NULL,
        PRIMARY KEY(W_ID)
)

CREATE TABLE TPCC_SCHEMA.DISTRICT(
        D_ID        SMALLINT NOT NULL,
        D_W_ID   SMALLINT 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_TAX     DECIMAL(4,4) NOT NULL,'96/04/18
        D_TAX     SMALLINT NOT NULL,
--      D_YTD     DECIMAL(12,2) NOT NULL,'98/11/24
        D_YTD     DECIMAL(12,0) NOT NULL,
        D_NEXT_O_ID INTEGER NOT NULL,
        PRIMARY KEY(D_W_ID,D_ID)
)

CREATE TABLE TPCC_SCHEMA.CUSTOMER(
--      C_ID        SMALLINT NOT NULL,'96/10/18
        C_ID        INTEGER   NOT NULL,
        C_D_ID    SMALLINT NOT NULL,
        C_W_ID    SMALLINT NOT NULL,
        C_FIRST   CHAR(16) NOT NULL,
        C_MIDDLE CHAR(2) NOT NULL,
        C_LAST    CHAR(16) NOT NULL,
        C_STREET_1 CHAR(20) NOT NULL,
        C_STREET_2 CHAR(20) NOT NULL,
        C_CITY    CHAR(20) NOT NULL,
        C_STATE CHAR(2)    NOT NULL,
        C_ZIP      CHAR(9)    NOT NULL,
        C_PHONE CHAR(16) NOT NULL,
--      C_SINCE   DECIMAL(14) NOT NULL,'96/04/18
        C_SINCE   CHAR(14) NOT NULL,
        C_CREDIT CHAR(2) NOT NULL,
--      C_CREDIT_LIM DECIMAL(12,2) NOT NULL,'98/11/24
        C_CREDIT_LIM DECIMAL(12,0) NOT NULL,
--      C_DISCOUNT DECIMAL(4,4) NOT NULL,'96/04/18
        C_DISCOUNT SMALLINT NOT NULL,
--      C_BALANCE DECIMAL(12,2) NOT NULL,    '98/11/24
        C_BALANCE DECIMAL(12,0) NOT NULL,
--      C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,'98/11/24
        C_YTD_PAYMENT DECIMAL(12,0) NOT NULL,
        C_PAYMENT_CNT SMALLINT NOT NULL,
        C_DELIVERY_CNT SMALLINT NOT NULL,
        C_DATA   CHAR(500) NOT NULL,
        PRIMARY KEY(C_W_ID, C_D_ID, C_ID)
)

CREATE TABLE TPCC_SCHEMA.ITEM(
        I_ID        INTEGER   NOT NULL,
        I_IM_ID   INTEGER   NOT NULL,
        I_NAME   CHAR(24) NOT NULL,
--      I_PRICE   DECIMAL(5,2) NOT NULL,'96/04/18
        I_PRICE   SMALLINT NOT NULL,
        I_DATA    CHAR(50) NOT NULL,
        PRIMARY KEY(I_ID)
)

CREATE TABLE TPCC_SCHEMA.STOCK(
        S_I_ID     INTEGER   NOT NULL,
        S_W_ID    SMALLINT NOT NULL,
        S_QUANTITY SMALLINT NOT NULL,
        S_DIST_01 CHAR(24) NOT NULL,
        S_DIST_02 CHAR(24) NOT NULL,
        S_DIST_03 CHAR(24) NOT NULL,
        S_DIST_04 CHAR(24) NOT NULL,
        S_DIST_05 CHAR(24) NOT NULL,
        S_DIST_06 CHAR(24) NOT NULL,
        S_DIST_07 CHAR(24) NOT NULL,
        S_DIST_08 CHAR(24) NOT NULL,
        S_DIST_09 CHAR(24) NOT NULL,
        S_DIST_10 CHAR(24) NOT NULL,
        S_YTD      INTEGER   NOT NULL,
        S_ORDER_CNT SMALLINT NOT NULL,
        S_REMOTE_CNT SMALLINT NOT NULL,
        S_DATA    CHAR(50) NOT NULL,
        PRIMARY KEY(S_W_ID, S_I_ID)
)

CREATE TABLE TPCC_SCHEMA.NEWORDER(
        NO_O_ID INTEGER   NOT NULL,
        NO_D_ID SMALLINT NOT NULL,
        NO_W_ID SMALLINT NOT NULL,
        PRIMARY KEY(NO_W_ID, NO_D_ID, NO_O_ID)
)

CREATE TABLE TPCC_SCHEMA.ORDERS(
        O_ID        INTEGER   NOT NULL,
```

```
        O_D_ID   SMALLINTNOT NULL,
        O_W_ID   SMALLINTNOT NULL,
--      O_C_ID   SMALLINTNOT NULL,'96/10/18
        O_C_ID   INTEGER   NOT NULL,
--      O_ENTRY_DDECIMAL(14)NOT NULL,'96/04/18
        O_ENTRY_DCHAR(14)NOT NULL,
        O_CARRIER_IDSMALLINT,
        O_OL_CNTSMALLINTNOT NULL,
        O_ALL_LOCALSMALLINTNOT NULL,
        PRIMARY KEY(O_W_ID, O_D_ID, O_ID)
)

CREATE TABLE TPCC_SCHEMA.ORDERLINE(
        OL_O_ID  INTEGER   NOT NULL,
        OL_D_ID  SMALLINTNOT NULL,
        OL_W_ID  SMALLINTNOT NULL,
        OL_NUMBERSMALLINTNOT NULL,
        OL_I_ID   INTEGER   NOT NULL,
        OL_SUPPLY_W_IDSMALLINTNOT NULL,
--      OL_DELIVERY_DDECIMAL(14),'96/04/18
        OL_DELIVERY_DCHAR(14),
        OL_QUANTITYSMALLINTNOT NULL,
--      OL_AMOUNTDECIMAL(6,2)NOT NULL,'96/04/18
        OL_AMOUNTINTEGERNOT NULL,
        OL_DIST_INFOCHAR(24)NOT NULL,
--      PRIMARY KEY(OL_W_ID, OL_D_ID, OL_O_ID, OL_NUMBER)
        PRIMARY KEY(OL_W_ID, OL_D_ID, OL_NUMBER, OL_O_ID )
)

CREATE TABLE TPCC_SCHEMA.HISTORY(
--      H_C_ID   SMALLINTNOT NULL,'96/10/18
        H_C_ID   INTEGER   NOT NULL,
        H_C_D_IDSMALLINTNOT NULL,
        H_C_W_IDSMALLINTNOT NULL,
        H_D_ID   SMALLINTNOT NULL,
        H_W_ID   SMALLINTNOT NULL,
--      H_DATE   DECIMAL(14)NOT NULL,'96/04/27
        H_DATE   CHAR(14)NOT NULL,
--      H_AMOUNTDECIMAL(6,2)NOT NULL,'96/04/18
        H_AMOUNTINTEGER NOT NULL,
        H_DATA   CHAR(24)NOT NULL
)
::::::::::::
ddl_db.mak
:::::::::::::
--
/*=================================================
==============*/
--/*                                          */
--
/*=================================================
==============*/
  CREATE DATABASE TPCC;

:::::::::::::
mkarc.sh
:::::::::::::
:
set -x
date

timex rdblog -G -a /DEV/rdsk/ARC1 2035M
sleep 1
```

```
timex rdblog -U -a /DEV/rdsk/ARC2
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC3
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC4
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC5
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC6
sleep 1
date

exit

timex rdblog -U -a /DEV/rdsk/ARC7
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC8
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC9
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC10
sleep 1
date

:::::::::::::
sh.rdbups
:::::::::::::
#!/bin/ksh

echo `date` Executing $0
set -v

rdbups -i TPCC.WAREHOUSE_1_DSI
rdbups -i TPCC.DISTRICT_1_DSI
rdbups -i TPCC.ITEM_1_DSI

rdbups -i TPCC.CUSTOMER_1_DSI
rdbups -i TPCC.CUSTOMER_X_1_DSI

rdbups -i TPCC.ORDERS_1_DSI
rdbups -i TPCC.ORDERS_IX_1_DSI
rdbups -i TPCC.NEWORDER_1_DSI
rdbups -i TPCC.NEWORDER_X_1_DSI
rdbups -i TPCC.HISTORY_1_DSI

rdbups -i TPCC.ORDERLIN_1_DSI

rdbups -i TPCC.STOCK_1_DSI

#wait
exit 0

:::::::::::::
sh.stored
:::::::::::::
:
set -x
#cd stored/Y_stored.PS.old
cd stored/Y_stored.PS
```

```
#Y_stored.sh
Y_stored_cent.sh


::::::::::::::
wttpccd1.c
::::::::::::::
/*
 *    File Name: wttpccd.ec
 *    Function Name: main()
 *            : item()
 *            : warehouse()
 *            : stock()
 *            : district()
 *            : customer()
 *            : history()
 *            : orders()
 *                      : new_order()
 *                      : make_address()
 *                      : lastname()
 *            : make_alpha_string()
 *            : make_number_string()
 *            : random_number()
 *            : set_seed()
 *            : nurand()
 *                      : init_permutation()
 *    Description: DB tpcc        , item,warehouse, stock,
 *             district, customer, history, orders, order_line,
 *             new_order        ,
 *                      (char )      .
 *    Author    :
 *    Reviewer:
 *
 *    COPYRIGHT FUJITSU Limited 1995
 *
 *        95-03-13
 *            -                   ( : %20s ==> %-s)
 *            -
 *             sprintf & fwrite   fprintf    (    )
 *            - ORDERS  ORDER_LINE         NULL
 *
 *        95-05-16
 *            - 10warehouse
 *        96-04-18
 *            - W-TAX, D-TAX, C-DISCOUNT, I-PRICE, OL-
 AMOUNT, H-AMOUNT
 *                      (DECIMAL -> SMALLINT or
 INTEGER)
 *            - C-SINCE, OL-DELIVERY-D, O-ENTRY-D
 *                      (DECIMAL -> CHAR)
 *        96-09-06
 *            - file
 *                 1.option       table      (3     parameter)
 *                 (0..all, 1..IT/ST/HI/CU,
 2..WH/DI/OL/OS/NO)
 *                 2.file
 *                 (/rdb/loaddata/[table ]/[warehouse    _
 ]
 *                 ex.: /rdb/loaddata/Customer/10_15 ..
 Customer Wh10-15)
 *            - Text    Binary   . DECIMAL
 *                      :)
 *                 <decimal(a,b)  format>
```

```
 *    decimal(10,2)   aa | aa | aa | aa | ab | bs (6byte=a/2+1)
 *        a=decimal        (b           )
 *        b=             (s  8bit      )
 *        s=      . (+)   "c", (-)   "d"
 *        ex.) +12345678.23 = '(0x) 01 | 23 | 45 | 67 | 82 | 3c
 *        program        FUNC.
 *
 *        "record   "              (
 *                  )
 *        ex.) w_ytd = -123.45;
 *                      :          :
 *                      :          :
 *                 ("record   " )
 *                 w_ytd_1 = 0x00;
 *                 w_ytd_2 = 0x00;
 *                 w_ytd_3 = 0x00;
 *                 w_ytd_4 = 0x00;
 *                 w_ytd_5 = 0x12;
 *                 w_ytd_6 = 0x34;
 *                 w_ytd_7 = 0x5d;
 *
 *    96-11-27
 *        - ol_i_id        (?)
 *            ol_i_id   1  10      n
 *            ( : n=2; 2,4,6,....99998,100000)
 *            n   setenv TPCRANDBY n
 *            n < TPCRANDBY < 10   or undefined     == 1
 *
 *    97-02-18
 *        - C_ID, H_C_ID, O_C_ID
 *                 (SMALLINT -> INTEGER)
 *        - I_IM_ID
 *
 *    97-02-18
 *        - fprintf -> sprintf + fwrite
 ( item,stock,customer,history,orders,orderline )
 *        - random_number  mac
 *        - make_alpha_string   make_number_string
 rund      .(              )
 *        - make_alpha_string                   .
 ( ORACLE , HP         )
 *        - get_permutation       ,o_c_id
 *        - TAB ID 221(c_last NURand    C)
 Issue      : C Value For NURand
 Specification : TPC-C,Clause 2.1.6
 c_last  NURand    C      .
 C-Load : DB            C
 C-Run :   (tran)                 C
 C-Delta: | C-Load - C-Run |
 *      C      ,[ 0,255]
 C-Delta       ,[65,119]    ,96,112
 C-Run  , 111            .
    Online : tranmain  Const
    Online : pptpcc2(shell)  u14i
 nurand()
  97-02-24
 *        - fopen + fwrite -> open + write
 ( item,stock,customer,history,orders,orderline )
  97-02-25
 *        - CUSTOMER,HISTORY
  97-02-27
 *        - sprintf   (typedef struct)
 - c_phone[16] -> c_phone[17]
```

```
*           97-03-05
*               - c_payment_cnt = 1 -> 0
*           97-03-11
*               - make_alpha_string          ,       .
*         ( ORACLE    )
*           98-11-06
*               -                  100
*               w_ytd, d_ytd, c_credit_lim, c_balance,
c_ytd_payment
*/

#include <stdio.h>
#include <string.h>
#include <sys/types.h>  /* 1994.12.28 add kawabata */
#include <sys/stat.h> /* 1995.02.24 add arakawa */
#include <fcntl.h> /* 1995.02.24 add arakawa */
#include <time.h>
#include <stdlib.h>/* 1996.11.27 added K.Fukui for "getenv()" */
#include <unistd.h>     /* 1997.02.24 write                */

/*#defineDBNAME   "tpcc"*/    /* DB   */
#define MAXITEMS100000/*         */
#define MAXSTOCK100000/* STOCK     */
#define DIST_PER_WARE   10/*              */
#define CUST_PER_DIST 3000/*              */
#define ORD_PER_DIST 3000/*               */
#define NEWORDS   900      /*                   */
#define CLS_CNT10000      /*                   */

#define CMT_CNT    3

#defineT256          16777216
#defineD256          65536
#defineNNUL_V        0x00
#defineNUL_V         0xFF

   /* 1997-02-27 sprintf    (typedef struct) */
typedef struct
  {
    chari_id_1,i_id_2,i_id_3,i_id_4 ;
    chari_im_id_1, i_im_id_2, i_im_id_3, i_im_id_4 ;
    char   i_name[24] ;
    chari_price_1, i_price_2 ;
    char   i_data[50] ;
  } item_str ;

typedef struct
  {
    chard_id_1,d_id_2;
    char   d_w_id_1,d_w_id_2;
    chard_name[10];
    chard_street_1[20];
    chard_street_2[20];
    chard_city[20];
    chard_state[2];
    chard_zip[9];
    chard_tax_1, d_tax_2;
    chard_ytd_1, d_ytd_2, d_ytd_3, d_ytd_4, d_ytd_5, d_ytd_6,
d_ytd_7;
    chard_next_o_id_1, d_next_o_id_2, d_next_o_id_3, d_next_o_id_4;
  } district_str ;

typedef struct
```

```
  {
    charw_id_1, w_id_2;
    charw_name[10];
    charw_street_1[20];
    charw_street_2[20];
    charw_city[20];
    charw_state[2];
    charw_zip[9];
    charw_tax_1, w_tax_2;
    charw_ytd_1, w_ytd_2, w_ytd_3, w_ytd_4, w_ytd_5, w_ytd_6, w_ytd_7;
  } warehouse_str ;

typedef struct
  {
    chars_i_id_1, s_i_id_2, s_i_id_3, s_i_id_4;
    chars_w_id_1, s_w_id_2;
    char   s_quantity_1, s_quantity_2;
    chars_dist_01[24];
    chars_dist_02[24];
    chars_dist_03[24];
    chars_dist_04[24];
    chars_dist_05[24];
    chars_dist_06[24];
    chars_dist_07[24];
    chars_dist_08[24];
    chars_dist_09[24];
    chars_dist_10[24];
    char   s_ytd_1, s_ytd_2, s_ytd_3, s_ytd_4;
    chars_order_cnt_1, s_order_cnt_2;
    char   s_remote_cnt_1, s_remote_cnt_2;
    chars_data[50];
  } stock_str ;

typedef struct
  {
    char   c_id_1, c_id_2, c_id_3, c_id_4;
    charc_d_id_1, c_d_id_2;
    charc_w_id_1, c_w_id_2;
    charc_first[16];
    charc_middle[2];
    charc_last[16];
    charc_street_1[20];
    charc_street_2[20];
    charc_city[20];
    charc_state[2];
    charc_zip[9];
    charc_phone[16];
    charc_since[14];
    charc_credit[2];
    charc_credit_lim_1, c_credit_lim_2, c_credit_lim_3, c_credit_lim_4;
    char   c_credit_lim_5, c_credit_lim_6, c_credit_lim_7;
    charc_discount_1, c_discount_2;
    charc_balance_1, c_balance_2, c_balance_3, c_balance_4;
    charc_balance_5, c_balance_6, c_balance_7;
    char   c_ytd_payment_1, c_ytd_payment_2, c_ytd_payment_3,
c_ytd_payment_4;
    charc_ytd_payment_5, c_ytd_payment_6, c_ytd_payment_7;
    charc_payment_cnt_1, c_payment_cnt_2;
    charc_delivery_cnt_1, c_delivery_cnt_2;
    charc_data[500];
  } customer_str ;

typedef struct
```

```
   {
     charh_c_id_1, h_c_id_2,h_c_id_3, h_c_id_4;
     charh_c_d_id_1, h_c_d_id_2;
     charh_c_w_id_1, h_c_w_id_2;
     charh_d_id_1, h_d_id_2;
     charh_w_id_1, h_w_id_2;
     charh_date[14];
     charh_amount_1, h_amount_2, h_amount_3, h_amount_4;
     char  h_data[24];
   } history_str ;

typedef struct
   {
     char  o_id_v1, o_id_v2;
     charo_id_1, o_id_2, o_id_3, o_id_4;
     char  o_d_id_v1, o_d_id_v2;
     charo_d_id_1, o_d_id_2;
     char  o_w_id_v1, o_w_id_v2;
     charo_w_id_1, o_w_id_2;
     char  o_c_id_v1, o_c_id_v2;
     charo_c_id_1, o_c_id_2, o_c_id_3, o_c_id_4;
     char  o_entry_d_v1,o_entry_d_v2;
     charo_entry_d[14];
     char  o_carrier_id_v1, o_carrier_id_v2;
     charo_carrier_id_1, o_carrier_id_2;
     char  o_ol_cnt_v1, o_ol_cnt_v2;
     charo_ol_cnt_1, o_ol_cnt_2;
     char  o_all_local_v1, o_all_local_v2;
     charo_all_local_1, o_all_local_2;
   } orders_str ;

typedef struct
   {
     char  ol_o_id_v1, ol_o_id_v2;
     char   ol_o_id_1, ol_o_id_2, ol_o_id_3, ol_o_id_4;
     char  ol_d_id_v1, ol_d_id_v2;
     charol_d_id_1, ol_d_id_2;
     char  ol_w_id_v1, ol_w_id_v2;
     charol_w_id_1, ol_w_id_2;
     char  ol_number_v1, ol_number_v2;
     char   ol_number_1, ol_number_2;
     char  ol_i_id_v1, ol_i_id_v2;
     charol_i_id_1, ol_i_id_2, ol_i_id_3, ol_i_id_4;
     char  ol_supply_w_id_v1, ol_supply_w_id_v2;
     charol_supply_w_id_1, ol_supply_w_id_2;
     char  ol_delivery_d_v1,ol_delivery_d_v2;
      char ol_delivery_d[14];
     char  ol_quantity_v1, ol_quantity_v2;
     charol_quantity_1, ol_quantity_2;
     char  ol_amount_v1, ol_amount_v2;
      char   ol_amount_1, ol_amount_2, ol_amount_3, ol_amount_4;
     char  ol_dist_info_v1,ol_dist_info_v2;
     charol_dist_info[24];
   } orderline_str ;

typedef struct
   {
     charno_o_id_1, no_o_id_2, no_o_id_3, no_o_id_4;
     charno_d_id_1, no_d_id_2;
     charno_w_id_1, no_w_id_2;
   } neworder_str ;

int len_i ;
```

```
      /* 1997-02-18 TAB ID 221(c_last NURand    C)              */
#define C_DELTA         87          /* | C_LOAD - C_RAN |       */
#define C_RUN           111         /* TRAN   NURand   C        */
#define C_LOAD     (C_DELTA+C_RUN)          /* DB LOAD NURand   C       */

      /* 1997-02-18  fprintf -> sprintf + fwrite            */
#define ITEM_SIZEsizeof(item_str)       /* ITEM             (84)*/
#define DISTRICT_SIZEsizeof(district_str)  /* DISTRICT          (98)*/
#define WAREHOUSE_SIZEsizeof(warehouse_str) /* WAREHOUSE        (92)*/
#define STOCK_SIZEsizeof(stock_str)     /* STOCK           (306)*/
#define CUSTOMER_SIZEsizeof(customer_str)  /* CUSTOMER         (672)*/
#define HISTORY_SIZEsizeof(history_str)   /* HISTORY          (54)*/
#define ORDERS_SIZE    sizeof(orders_str)    /* ORDERS       (32+16)*/
#define ORDERLINE_SIZEsizeof(orderline_str) /* ORDERLINE      (60+20)*/
#define NEWORDER_SIZE sizeof(neworder_str)  /* HISTORY          ( 8)*/

#define ITEM_COUNT     1024            /* ITEM               */
#define STOCK_COUNT    1024            /* STOCK              */
#define CUSTOMER_COUNT    512          /* CUSTOMER            */
#define HISTORY_COUNT    1024          /* HISTORY            */
#define ORDERS_COUNT     1024          /* ORDERS            */
#define ORDERLINE_COUNT  2048          /* ORDERLINE           */

/*            */
/*            */
char  yyyymmddhhmmss[15];

/*            . ..._1 ..._7  96-09-06           */
/*      (INTEGER:_1 _4 / SMALLINT:_1 _2 / DECIMAL:_1 _7) */

int   i_id;
int        i_id_1, i_id_2, i_id_3, i_id_4;
int   i_im_id;                                  /* 97-02-18    */
int        i_im_id_1, i_im_id_2, i_im_id_3, i_im_id_4;/* 97-02-18    */
char i_name[25];
int   i_price;
int        i_price_1, i_price_2;
char  i_data[51];

short w_id;
int        w_id_1, w_id_2;
char  w_name[11];
char  w_street_1[21];
char  w_street_2[21];
char  w_city[21];
char  w_state[3];
char  w_zip[10];
int   w_tax;
int        w_tax_1, w_tax_2;
float w_ytd;
int        w_ytd_1, w_ytd_2, w_ytd_3, w_ytd_4, w_ytd_5, w_ytd_6, w_ytd_7;

int   s_i_id;
int        s_i_id_1, s_i_id_2, s_i_id_3, s_i_id_4;
short s_w_id;
int        s_w_id_1, s_w_id_2;
int   s_quantity;
int        s_quantity_1, s_quantity_2;
char  s_dist_01[25];
char  s_dist_02[25];
char  s_dist_03[25];
char  s_dist_04[25];
char  s_dist_05[25];
```

```
char s_dist_06[25];
char s_dist_07[25];
char s_dist_08[25];
char s_dist_09[25];
char s_dist_10[25];
int        s_ytd;
int        s_ytd_1, s_ytd_2, s_ytd_3, s_ytd_4;
int    s_order_cnt;
int        s_order_cnt_1, s_order_cnt_2;
int    s_remote_cnt;
int        s_remote_cnt_1, s_remote_cnt_2;
char s_data[51];

short d_id;
int        d_id_1,d_id_2;
short d_w_id;
int        d_w_id_1,d_w_id_2;
char d_name[11];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
int        d_tax;
int        d_tax_1, d_tax_2;
char work[10];
float d_ytd;
int        d_ytd_1, d_ytd_2, d_ytd_3, d_ytd_4, d_ytd_5, d_ytd_6,
d_ytd_7;
int    d_next_o_id;
int        d_next_o_id_1, d_next_o_id_2, d_next_o_id_3,
d_next_o_id_4;

int        c_id;                            /* 97-02-18
short -> int*/
int        c_id_1, c_id_2, c_id_3, c_id_4;/* 97-02-18 3  4       */
short c_d_id;
int        c_d_id_1, c_d_id_2;
short c_w_id;
int        c_w_id_1, c_w_id_2;
char c_first[17];
char c_middle[3];
char c_last[17];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];      /* 1997.02.27 */
char c_since[15];
char c_credit[3];
float c_credit_lim;
int        c_credit_lim_1, c_credit_lim_2, c_credit_lim_3,
c_credit_lim_4;
int        c_credit_lim_5, c_credit_lim_6, c_credit_lim_7;
int    c_discount;
int        c_discount_1, c_discount_2;
float c_balance;
int        c_balance_1, c_balance_2, c_balance_3, c_balance_4;
int        c_balance_5, c_balance_6, c_balance_7;
float c_ytd_payment;
int        c_ytd_payment_1, c_ytd_payment_2, c_ytd_payment_3,
c_ytd_payment_4;
```

```
int        c_ytd_payment_5, c_ytd_payment_6, c_ytd_payment_7;
int    c_payment_cnt;
int        c_payment_cnt_1, c_payment_cnt_2;
int    c_delivery_cnt;
int        c_delivery_cnt_1, c_delivery_cnt_2;
char c_data[501];

int        h_c_id;                            /* 97-02-18       short -> int*/
int        h_c_id_1, h_c_id_2,h_c_id_3, h_c_id_4;/* 97-02-18 3  4      */
short h_c_d_id;
int        h_c_d_id_1, h_c_d_id_2;
short h_c_w_id;
int        h_c_w_id_1, h_c_w_id_2;
short h_d_id;
int        h_d_id_1, h_d_id_2;
short h_w_id;
int        h_w_id_1, h_w_id_2;
char h_date[15];
int    h_amount;
int        h_amount_1, h_amount_2, h_amount_3, h_amount_4;
char   h_data[25];

int    o_id;
int        o_id_1, o_id_2, o_id_3, o_id_4;
short o_d_id;
int        o_d_id_1, o_d_id_2;
short o_w_id;
int        o_w_id_1, o_w_id_2;
int    o_c_id;                            /* 97-02-18        short -> int*/
int        o_c_id_1, o_c_id_2, o_c_id_3, o_c_id_4;/* 97-02-18 3  4      */
char o_entry_d[15];
short o_carrier_id;
int        o_carrier_id_1, o_carrier_id_2;
short o_ol_cnt;
int        o_ol_cnt_1, o_ol_cnt_2;
short o_all_local;
int        o_all_local_1, o_all_local_2;

int    ol_o_id;
int        ol_o_id_1, ol_o_id_2, ol_o_id_3, ol_o_id_4;
short ol_d_id;
int        ol_d_id_1, ol_d_id_2;
short ol_w_id;
int        ol_w_id_1, ol_w_id_2;
short ol_number;
int        ol_number_1, ol_number_2;
int    ol_i_id;
int        ol_i_id_1, ol_i_id_2, ol_i_id_3, ol_i_id_4;
short  ol_supply_w_id;
int        ol_supply_w_id_1, ol_supply_w_id_2;
char ol_delivery_d[15];
int    ol_quantity;
int        ol_quantity_1, ol_quantity_2;
int    ol_amount;
int        ol_amount_1, ol_amount_2, ol_amount_3, ol_amount_4;
char ol_dist_info[25];

int    no_o_id;
int        no_o_id_1, no_o_id_2, no_o_id_3, no_o_id_4;
short no_d_id;
int        no_d_id_1, no_d_id_2;
short no_w_id;
int        no_w_id_1, no_w_id_2;
```

```
/*shortc;        */    /* NURand          */
short ocid[CUST_PER_DIST];/* o_c_id        */
short counter;    /* o_c_id          */

/*   :961127:K.Fukui: I_ID       (main    ) */
     char *EnvGetI_ID;
     int       I_ID_Rand_by;
/*   :961127:K.Fukui: (above is all) */

void  item();
void  warehouse();
void  stock();
void  district();
void  customer();
void  history();       /* 1997.02.25    */
void  orders();
void  make_address();
void  lastname();
int       make_alpha_string();
int       make_number_string();
#ifdef call_rand
int       random_number();
#else
#define random_number(x,y)    ((int)(lrand48()%(y-x+1)) + x)
#endif
void  set_seed();
int       nurand();
void  init_permutation();
/*int get_permutation();*/

/*            */
FILE *fst1;
FILE *fst2;
FILE *fst3;
FILE *fst4;
FILE *fst5;
FILE *fst6;
FILE *fst7;
FILE *fst8;
FILE *fst9;
int    wst;
int    op_item    ;        /* 1997.02.24 open+write */
int    op_stock   ;        /* 1997.02.24 open+write */
int    op_customer  ;       /* 1997.02.24 open+write */
int    op_history  ;       /* 1997.02.24 open+write */
int    op_orders   ;       /* 1997.02.24 open+write */
int    op_orderline ;      /* 1997.02.24 open+write */

char  fileout[100];/*          */
char  filedum[100];

/*
 *     Function: main()
 *     Description: DB          , item, warehouse       ,
 *                        .
 *     Parameters: 1. argc,
 *                       2. argv,
 *
 *
 *     Grobals Ref: nothing
 *     Grobals Out: 1. yyyymmddhhmmss,
 *     Returns    : 0
```

```
*               1
 */

int
main(argc, argv)
int    argc;
char **argv;
{
     time_t      tod;  /*              */
     struct tm*stm;/*            */
     int    count_ware;
     int    last_ware;
     int    base_ware;
     int make_type;
     int    mk_loop;
     char  sw_buf[1];

     /*              */
     if (argc < 5) {
         printf("usage: wttpcc [output_dir] "
         "[start_warehouse] "
         "[end_warehouse] "
         "[maketype]..\n\n");
         printf(" [maketype] make data seeds for rdbsloader "
         "(multiple designation available)\n");
         printf("        I:Item, D:District, W:Warehouse, S:Stock,\n");
         printf("        C:Customer, H:History,"
             " O:Orders/OrderLine/NewOrder\n");
         exit(1);
     }

     /*              */
     strcpy( fileout, argv[1] );
     base_ware = atoi(argv[2]);
     last_ware = atoi(argv[3]);
/* make_type = atoi(argv[3]); */
     count_ware = last_ware - base_ware;
     /*if (count_ware <= 0) {   1997.02.21 */
     if (count_ware < 0) {
         printf("%s: invalid warehouse count\n",argv[0]);
         exit(1);
     }

/* system("date"); */
/* printf("%s start\n",argv[0]); */
     printf("wttpcc:    :          : (%d-%dWH) ",base_ware,last_ware);
     for( mk_loop = 4; mk_loop <= argc-1; mk_loop++ ){

         strcpy(sw_buf, argv[mk_loop]);

         switch( sw_buf[0] ){
             case 'I':   printf("Item(%3d), ",ITEM_SIZE);
             break;
             case 'W':   printf("Warehouse(%3d), ",WAREHOUSE_SIZE);
             break;
             case 'S':   printf("Stock(%3d), ",STOCK_SIZE);
             break;
             case 'D':   printf("District(%3d), ",DISTRICT_SIZE);
             break;
             case 'C':   printf("Customer(%3d), ",CUSTOMER_SIZE);
             break;
             case 'H':   printf("History(%3d), ",HISTORY_SIZE);
```

```
            break;
            case 'O':   printf("Orders(%3d), N.Order(%3d),
O.Line(%3d), ",

ORDERS_SIZE,NEWORDER_SIZE,ORDERLINE_SIZE);
            break;
        }
    }
    printf("to %s\n", fileout);

    /*    :961127:K.Fukui: ITEM_ID        (?)(    1 10   n
            I_ID              )

            "TPCRANDBY"   integer n (100000/n  NUrand
,
            *n   )          .
        ( 1 < TPCRANDBY < MAXITEMS           "1"      )
*/


    EnvGetI_ID = getenv( "TPCRANDBY" );

    if( EnvGetI_ID == NULL ){
        I_ID_Rand_by = 1;
        printf("wttpcc:    : "
    "TPCRANDBY:                OL_I_ID          \n");

    } else {

        I_ID_Rand_by = atoi( EnvGetI_ID );
        printf("wttpcc:    : "
        "TPCRANDBY: %d        OL_I_ID          \n",
I_ID_Rand_by);

        if( ( I_ID_Rand_by < 1 ) || ( I_ID_Rand_by > MAXITEMS
) ){

            I_ID_Rand_by = 1;
            printf("wttpcc:    : "
        "TPCRANDBY:         (1  %d)."
        "   OL_I_ID          \n",
        MAXITEMS);

        }

    }

    /*    :961127:K.Fukui: (above is all) */

    /*              */
    set_seed(time(0));

    /* 1997-02-18 TAB ID 221(c_last NURand    C)        */
    /* NURand              */
    /* c = random_number(0, 255);   */
    printf("wttpcc:   : TAB ID 221    C-Delta = %d \n",C_DELTA) ;
printf("       C-Load  NURAND     C = %d \n",C_LOAD ) ;
printf("       C-Run   NURAND     C = %d \n",C_RUN  ) ;

    /*              */
    time(&tod);
    stm = localtime(&tod);
    sprintf(yyyymmddhhmmss,"%04d%02d%02d%02d%02d%02d",
            stm->tm_year+1900,stm->tm_mon+1,stm->tm_mday,
```

```
            stm->tm_hour,stm->tm_min,stm->tm_sec);

for( mk_loop = 4; mk_loop <= argc-1; mk_loop++ ){

    strcpy(sw_buf, argv[mk_loop]);

    switch( sw_buf[0] ){
        case 'I':
            if ( base_ware == 1 )
        {
                fprintf(stderr,"wttpcc:    : ITEM           \n");
                /* item            */
                item();
                fprintf(stderr,"wttpcc:    : ITEM             \n");
            }
    else
    {
      printf("wttpcc:    : "
        "   warehouse '1'        ,"
        "ITEM            \n") ;
    }
            break;

        case 'W':
            fprintf(stderr,"wttpcc:    : "
    "WAREHOUSE (%d  %dwh)            \n",
    base_ware,last_ware);
            /* warehouse            */
            warehouse(base_ware,last_ware);
            fprintf(stderr,"wttpcc:    : "
    "WAREHOUSE (%d  %dwh)            \n",
    base_ware,last_ware);
            break;

        case 'S':
            fprintf(stderr,"wttpcc:    : "
    "STOCK (%d  %dwh)            \n",
    base_ware, last_ware);
            /* stock            */
            stock(base_ware,last_ware);
            fprintf(stderr,"wttpcc:    : "
    "STOCK (%d  %dwh)            \n",
    base_ware, last_ware);
            break;

        case 'D':
            fprintf(stderr,"wttpcc:    : "
    "DISTRICT (%d  %dwh)           \n",
    base_ware, last_ware);
            /* district            */
            district(base_ware,last_ware);
            fprintf(stderr,"wttpcc:    : "
    "DISTRICT (%d  %dwh)            \n",
    base_ware, last_ware);
            break;

        case 'C':
            fprintf(stderr,"wttpcc:    : "
    "CUSTOMER (%d  %dwh)            \n",
    base_ware, last_ware);
            /* customer            */
            customer(base_ware,last_ware);
            fprintf(stderr,"wttpcc:    : "
```

```
            "CUSTOMER (%d  %dwh)                  \n",
            base_ware, last_ware);
                    break;

              case 'H':
                    fprintf(stderr,"wttpcc:    : "
            "HISTORY (%d  %dwh)              \n",
            base_ware, last_ware);
                    /* history         */
                    history(base_ware,last_ware);
                    fprintf(stderr,"wttpcc:    : "
            "HISTORY (%d  %dwh)              \n",
            base_ware, last_ware);
                    break;

              case 'O':
                    fprintf(stderr,"wttpcc:    : "
            "ORDERS/O.LINE/N.ORDER (%d  %dwh)
\n",
            base_ware, last_ware);
                    /* orders          */
                    orders(base_ware,last_ware);
                    fprintf(stderr,"wttpcc:    : "
            "ORDERS/O.LINE/N.ORDER (%d  %dwh)
\n",
            base_ware, last_ware);
          }
      }

/*    system("date"); */

      /*              */
      return(0);
}

/*
 *    Function: item()
 *    Description: item                .
 *    Parameters: nothing
 *    Grobals Ref: nothing
 *    Grobals Out: nothing
 *    Returns    : nothing
 */

void
item()
{
      short       idatasiz;
      short       orig[MAXITEMS];
      int         pos;
      int         cnt;
      long        d_100 = 100.0;
      /* 1997-02-18  fprintf -> sprintf + fwrite      */
      int     item_lpcnt              ; /*          */
      char    *item_ap                ; /*          */
      item_str  *item_cp              ; /*          */

      /*          */
      sprintf( filedum, "%s/data", fileout );

      /*if ((fst1 = fopen( filedum , "w"))==NULL){   1997.02.24   */
      if ((op_item = open( filedum ,
O_WRONLY|O_CREAT|O_TRUNC,
            S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH ))==NULL){
            printf("wttpcc:    : %s:           \n", filedum);
            exit(1);
        }

    /* 1997-02-18  fprintf -> sprintf + fwrite      */
/* ITEM                      */
item_ap = (char *)malloc((size_t)ITEM_SIZE*ITEM_COUNT);
if ( item_ap == NULL )            /*              */
 {                                /*              */
    printf("Malloc failed.(item)\n")        ; /*              */
    exit(1)                       ; /*              */
 }                                /*              */
item_cp    = (item_str *)item_ap       ; /*              */
item_lpcnt = 0                    ; /*              */

    /*   orig  MAXITEMS        , i_data "ORIGINAL"
        10                    */
    memset(orig, 0, sizeof(orig));
    for (cnt = 0; cnt < (MAXITEMS / 10); cnt++) {
        do {
            pos = random_number(1, MAXITEMS);
        } while (orig[pos - 1]);
        orig[pos - 1] = 1;
    }

    /* i_id 1-MAXITEMS          , MAXITEMS   */
    /* item                    */
    for (i_id = 1; i_id <= MAXITEMS; i_id++) {

        /* i_name         */
        make_alpha_string(14, 24, i_name);

        /* i_data      ,   10%       ORIGINAL      */
        idatasiz = make_alpha_string(26, 50, i_data);
        if (orig[i_id - 1]) {
            pos = random_number(0, idatasiz - 8);
            strncpy(&i_data[pos], "ORIGINAL", 8);
        }

        /* record    :    : 1997-02-27 */
memset(item_cp->i_name, ' ' ,24)     ;
len_i = strlen(i_name)               ;
strncpy(item_cp->i_name,i_name,len_i) ;
memset(item_cp->i_data, ' ' ,50)     ;
len_i = strlen(i_data)               ;
strncpy(item_cp->i_data,i_data,len_i) ;

        /* record    :    : 96/09/06     */
item_cp->i_id_1 = i_id / T256;
        item_cp->i_id_2 = (i_id - ( i_id_1 * T256 )) / D256;
        item_cp->i_id_3 = (i_id - ( i_id_1 * T256 )
                - ( i_id_2 * D256 )) / 256 ;
        item_cp->i_id_4 = i_id % T256 ;

    /* i_im_id     : 97-02-18 start */
        i_im_id = random_number(1, 10000);
        item_cp->i_im_id_1 = i_im_id / T256;
        item_cp->i_im_id_2 = (i_im_id - (i_im_id_1 * T256)) / D256;
        item_cp->i_im_id_3 = (i_im_id - (i_im_id_1 * T256)
                    - (i_im_id_2 * D256)) / 256;
        item_cp->i_im_id_4 = i_im_id % T256;
        /* i_im_id     : 97-02-18 end */
```

```
        /* i_price        */
    /* i_price /= d_100; */
        i_price = random_number(100, 10000);
        item_cp->i_price_1 = i_price / 256 ;
        item_cp->i_price_2 = i_price % 256 ;

    item_cp   = item_cp   + 1 ;
    item_lpcnt = item_lpcnt + 1 ;

    if ( item_lpcnt == ITEM_COUNT )
     {
     write(op_item,
         item_ap,
         (size_t)ITEM_SIZE * (size_t)ITEM_COUNT ) ;
     item_cp   = (item_str *)item_ap     ;
     item_lpcnt = 0            ;
     }
    }

    /* 1997-02-18  fprintf -> sprintf + fwrite   */
    if ( item_lpcnt != 0 )
     {
     write(op_item,
         item_ap,
         (size_t)ITEM_SIZE * (size_t)item_lpcnt ) ;
     }

    /*             */
    close(op_item);

  /*             */
  free(item_ap) ;

    /*             */
    return;
}

/*
 *    Function: warehouse()
 *    Description: warehouse            .
 *    Parameters: 1. base_ware,
 *    Parameters: 2. last_ware,
 *    Grobals Ref: nothing
 *    Returns    : nothing
 */

void
warehouse(base_ware,last_ware)
int    base_ware;
int    last_ware;
{

    /*      */
    int        filecount = 1;
    int        outfilecount;
    char       filename[64];

    long       d_10000 = 10000.0;
/*   w_ytd = 300000.00; record            */
    w_ytd = 30000000;/* 98-11-06 */
    outfilecount = ((base_ware-1)/10) + 1;
```

```
/*             */
sprintf(filename , "%s/WH%d_%d", fileout, base_ware, last_ware);
if ((fst2 = fopen(filename , "w"))==NULL){
    printf("wttpcc:     : %s:            \n",filename);
    exit(1);
}

/* w_id              , count_ware      */
/* warehouse                     */
for (w_id = base_ware; w_id <= last_ware; w_id++) {

    /* w_name        */
    make_alpha_string(6, 10, w_name);

    /*              */
    make_address(w_street_1, w_street_2, w_city, w_state, w_zip);

/* w_tax /= d_10000;*/
    w_tax = random_number(0, 2000);

    /* record    :    : 96/09/06 fukui */

    w_id_1 = w_id / 256;
    w_id_2 = w_id % 256;
    w_tax_1 = w_tax / 256;
    w_tax_2 = w_tax % 256;
    w_ytd_1 = 0x00;/* w_ytd: +300000.00 */
    w_ytd_2 = 0x00;
    w_ytd_3 = 0x03;
    w_ytd_4 = 0x00;
    w_ytd_5 = 0x00;
    w_ytd_6 = 0x00;
    w_ytd_7 = 0x0c;

    fprintf(fst2 ,
        "%c%c"
        "%-10s"
        "%-20s"
        "%-20s"
        "%-20s"
        "%-2s"
        "%-9s"
        "%c%c"
        "%c%c%c%c%c%c%c",
        w_id_1,w_id_2,
        w_name,
        w_street_1,w_street_2,
        w_city,
        w_state,
        w_zip,
        w_tax_1,w_tax_2,
        w_ytd_1,w_ytd_2,w_ytd_3,w_ytd_4,w_ytd_5,w_ytd_6,w_ytd_7);

    filecount++;

}

/*             */
fclose(fst2);

/*             */
return;
```

```
}

/*
 *      Function: stock()
 *      Description: stock          .
 *      Parameters: 1. base_ware,
 *      Parameters: 2. last_ware,
 *      Grobals Ref: nothing
 *      Grobals Out: nothing
 *      Returns    : nothing
 */

void
stock(base_ware,last_ware)
int    base_ware;
int    last_ware;
{
        /*       */
        short sdatasiz;
        short orig[MAXITEMS];
        int         pos;
        int         cnt;
        int         filecount = 1;
        int         outfilecount;
        char  filename[64];
        /* 1997-02-18  fprintf -> sprintf + fwrite     */
    int        stock_lpcnt              ; /*             */
    char       *stock_ap                ; /*             */
    stock_str  *stock_cp                ; /*             */

        s_ytd = 0;
        s_order_cnt = 0;
        s_remote_cnt = 0;
        outfilecount = ((base_ware-1)/10) + 1;

        /*                  */
        sprintf(filename , "%s/ST%d_%d" ,fileout, base_ware,
last_ware);
        /*if ((fst3 = fopen(filename , "w"))==NULL){     1997.02.24 */
        if ((op_stock = open( filename ,
O_WRONLY|O_CREAT|O_TRUNC,
             S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH
))==NULL){
             printf("wttpcc:      : %s:           \n" , filename);
             exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite     */
    /* STOCK                         */
    stock_ap = (char *)malloc((size_t)STOCK_SIZE*STOCK_COUNT);
    if ( stock_ap == NULL )           /*             */
     {                      /*             */
       printf("Malloc failed.(stock)\n")     ; /*             */
       exit(1)                        ; /*             */
     }                      /*             */
    stock_cp   = (stock_str *)stock_ap       ; /*             */
    stock_lpcnt = 0                    ; /*             */

        /* w_id          count_ware           */
        for (s_w_id = base_ware; s_w_id <= last_ware; s_w_id++){

             fprintf(stderr,"wttpcc:     : "
                 "STOCK %d/%d  %d       \n" ,
```

```
                 s_w_id, base_ware, last_ware);

    /*    orig  MAXSTOCK
10                              */
    memset(orig, 0, sizeof(orig));
    for (cnt = 0; cnt < (MAXSTOCK / 10); cnt++) {
          do {
               pos = random_number(1, MAXSTOCK);
          } while (orig[pos - 1]);
          orig[pos - 1] = 1;
    }

    /* s_i_id              , MAXSTOCK    */
    /* stock                            */
    for (s_i_id = 1; s_i_id <= MAXSTOCK; s_i_id++) {

          /* s_quantity          */
          s_quantity = random_number(10, 100);

          /* s_dist_01 .. s_dist_10        */
          make_alpha_string(24, 24, s_dist_01);
          make_alpha_string(24, 24, s_dist_02);
          make_alpha_string(24, 24, s_dist_03);
          make_alpha_string(24, 24, s_dist_04);
          make_alpha_string(24, 24, s_dist_05);
          make_alpha_string(24, 24, s_dist_06);
          make_alpha_string(24, 24, s_dist_07);
          make_alpha_string(24, 24, s_dist_08);
          make_alpha_string(24, 24, s_dist_09);
          make_alpha_string(24, 24, s_dist_10);

          /* s_data        ,    10%      ORIGINAL       */
          sdatasiz = make_alpha_string(26, 50, s_data);
          if (orig[s_i_id - 1]) {
               pos = random_number(0, sdatasiz - 8);
               strncpy(&s_data[pos], "ORIGINAL", 8);
          }

          /* record   :    : 1997-02-27 */
strncpy(stock_cp->s_dist_01,s_dist_01,24) ;
strncpy(stock_cp->s_dist_02,s_dist_02,24) ;
strncpy(stock_cp->s_dist_03,s_dist_03,24) ;
strncpy(stock_cp->s_dist_04,s_dist_04,24) ;
strncpy(stock_cp->s_dist_05,s_dist_05,24) ;
strncpy(stock_cp->s_dist_06,s_dist_06,24) ;
strncpy(stock_cp->s_dist_07,s_dist_07,24) ;
strncpy(stock_cp->s_dist_08,s_dist_08,24) ;
strncpy(stock_cp->s_dist_09,s_dist_09,24) ;
strncpy(stock_cp->s_dist_10,s_dist_10,24) ;

memset(stock_cp->s_data,' ',50)           ;
len_i = strlen(s_data)               ;
strncpy(stock_cp->s_data,s_data,len_i)    ;

          /* record   :    : 96/09/09 fukui */
          stock_cp->s_i_id_1 = s_i_id / T256;
          stock_cp->s_i_id_2 = (s_i_id - (s_i_id_1 * T256)) / D256;
          stock_cp->s_i_id_3 = (s_i_id - (s_i_id_1 * T256)
                    - (s_i_id_2 * D256)) / 256;
          stock_cp->s_i_id_4 = s_i_id % T256;
          stock_cp->s_w_id_1 = s_w_id / 256;
          stock_cp->s_w_id_2 = s_w_id % 256;
          stock_cp->s_quantity_1 = s_quantity / 256;
```

```
                stock_cp->s_quantity_2 = s_quantity % 256;
                stock_cp->s_ytd_1 = s_ytd / T256;
                stock_cp->s_ytd_2 = (s_ytd - (s_ytd_1*T256)) / D256;
                stock_cp->s_ytd_3 = (s_ytd - (s_ytd_1*T256)-
(s_ytd_2*D256)) / 256;
                stock_cp->s_ytd_4 = s_ytd % T256;
                stock_cp->s_order_cnt_1 = s_order_cnt / 256;
                stock_cp->s_order_cnt_2 = s_order_cnt % 256;
                stock_cp->s_remote_cnt_1 = s_remote_cnt / 256;
                stock_cp->s_remote_cnt_2 = s_remote_cnt % 256;

        stock_cp   = stock_cp   + 1 ;
        stock_lpcnt = stock_lpcnt + 1 ;

        if ( stock_lpcnt == STOCK_COUNT )
          {
           write(op_stock,
                stock_ap,
                (size_t)STOCK_SIZE * (size_t)STOCK_COUNT ) ;
            stock_cp   = (stock_str *)stock_ap    ;
            stock_lpcnt = 0           ;
          }
            }

            filecount++;
      }

      /* 1997-02-18  fprintf -> sprintf + fwrite   */
      if ( stock_lpcnt != 0 )
       {
        write(op_stock,
              stock_ap,
              (size_t)STOCK_SIZE * (size_t)stock_lpcnt ) ;
       }

        /*              */
        close(op_stock);

    /*              */
    free(stock_ap) ;

        /*              */
        return;

}

/*
 *      Function: district()
 *      Description: district         .
 *      Parameters: 1. base_ware,
 *      Parameters: 2. last_ware,
 *      Grobals Ref: nothing
 *      Grobals Out: nothing
 *      Returns    : nothing
 */

void
district(base_ware,last_ware)
int    base_ware;
int    last_ware;
{
        /*     */
        long  d_10000 = 10000.0;
```

```
int       filecount = 1;
int       outfilecount;
char  filename[64];

/*   d_ytd = 30000.00; record          */
d_ytd = 3000000;/* 98-11-06 */
d_next_o_id = 3001;
outfilecount = ((base_ware-1)/10) +1;

/*              */
sprintf(filename , "%s/DI%d_%d" ,fileout, base_ware, last_ware);
if ((fst4 = fopen(filename , "w"))==NULL){
    printf("wttpcc:     : %s:           \n" , filename);
    exit(1);
}

/* w_id         , count_ware       */
for (d_w_id = base_ware; d_w_id <= last_ware; d_w_id++){

   /* d_id           , DIST_PER_WARE       */
   for (d_id = 1; d_id <= DIST_PER_WARE; d_id++) {

        /* d_name        */
        make_alpha_string(6, 10, d_name);

        /*              */
        make_address(d_street_1, d_street_2, d_city, d_state, d_zip);

        /* d_tax        */
  /* d_tax /= d_10000;*/
        d_tax = random_number(0, 2000);

        /* record   :   : 96-09-06 fukui */
        d_id_1 = d_id / 256;
        d_id_2 = d_id % 256;
        d_w_id_1 = d_w_id / 256;
        d_w_id_2 = d_w_id % 256;
        d_tax_1 = d_tax / 256;
        d_tax_2 = d_tax % 256;
        d_ytd_1 = 0x00;/* d_ytd: 30000.00 */
        d_ytd_2 = 0x00;
        d_ytd_3 = 0x00;
        d_ytd_4 = 0x30;
        d_ytd_5 = 0x00;
        d_ytd_6 = 0x00;
        d_ytd_7 = 0x0c;
        d_next_o_id_1 = d_next_o_id / T256;
        d_next_o_id_2 = (d_next_o_id-(d_next_o_id_1*T256))/D256;
        d_next_o_id_3 = (d_next_o_id-(d_next_o_id_1*T256)
                    -(d_next_o_id_2*D256))/256;
        d_next_o_id_4 = d_next_o_id % T256;

        fprintf(fst4 ,"%c%c"
          "%c%c"
          "%-10s"
          "%-20s%-20s%-20s%-2s%-9s"
          "%c%c"
          "%c%c%c%c%c%c%c"
          "%c%c%c%c",
        d_id_1,d_id_2,
        d_w_id_1,d_w_id_2,
        d_name,
        d_street_1,d_street_2,d_city,d_state,d_zip,
```

```
                d_tax_1,d_tax_2,

d_ytd_1,d_ytd_2,d_ytd_3,d_ytd_4,d_ytd_5,d_ytd_6,d_ytd_7,

d_next_o_id_1,d_next_o_id_2,d_next_o_id_3,d_next_o_id_4);


            }
            filecount++;
        }

        /*              */
        fclose(fst4);

        /*              */
        return;

}

/*
 *      Function: customer()
 *      Description: customer           .
 *      Parameters: 1. base_ware,
 *      Parameters: 2. last_ware,
 *      Grobals Ref: yyyymmddhhmmss,
 *      Grobals Out: nothing
 *      Returns   : nothing
 */

void
customer(base_ware,last_ware)
int     base_ware;
int     last_ware;
{
        /*      */
        long        d_10000 = 1000.0;
        long        d_100 = 100.0;
        int             filecount = 1;
        int             outfilecount;
        char        filename1[64];
        /* 1997-02-18  fprintf -> sprintf + fwrite    */
    int         customer_lpcnt          ; /*              */
    char        *customer_ap            ; /*              */
    customer_str *customer_cp           ; /*              */

/*    c_credit_lim = 50000.00; record      ,        */
      c_credit_lim = 5000000; /* 1998.11.06 */
/*    c_balance = -10.00; record      ,        */
      c_balance = -1000;/* 1998.11.06  */
/*    c_ytd_payment = 10.00; record      ,        */
      c_ytd_payment = 1000;/* 1998.11.06 */
      c_payment_cnt = 0;       /* 1997.03.05           */
      c_delivery_cnt = 0;
      strcpy(c_middle, "OE");
      strcpy(c_since, yyyymmddhhmmss);

      outfilecount = ((base_ware-1)/10) +1;
      /*              */
      sprintf(filename1 , "%s/CU%d_%d" ,fileout, base_ware,
last_ware);
      /*if ((fst5 = fopen(filename1 , "w"))==NULL){  1997.02.24 */
      if ((op_customer = open(
filename1,O_WRONLY|O_CREAT|O_TRUNC,
```

```
              S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH ))==NULL){
        printf("wttpcc:       : %s:               \n" , filename1);
        exit(1);
    }

    /* 1997-02-18  fprintf -> sprintf + fwrite     */
  /* CUSTOMER                            */
  customer_ap = (char *)malloc((size_t)(CUSTOMER_SIZE*CUSTOMER_COUNT)
) ;
    if ( customer_ap == NULL )               /*                  */
    {                                    /*              */
      printf("Malloc failed.(customer)\n")     ; /*              */
      exit(1)                             ; /*              */
    }                                     /*              */
  customer_cp   = (customer_str *)customer_ap  ; /*              */
  customer_lpcnt = 0                     ; /*              */

    /* w_id          count_ware          */
    for (c_w_id = base_ware; c_w_id <= last_ware; c_w_id++){

        fprintf(stderr,"wttpcc:      : CUSTOMER %d/%d  %d      \n"
                              ,c_w_id, base_ware,last_ware);

        /* d_id            DIST_PER_WARE          */
        for (c_d_id = 1; c_d_id <= DIST_PER_WARE; c_d_id++){

            /* c_id            CUST_PER_DIST         */
            /* coustomer, history                     */
            for (c_id = 1; c_id <= CUST_PER_DIST; c_id++) {

                make_alpha_string(8, 16, c_first)    ;

            /* 1997-02-18 TAB ID 221(c_last NURand     C)         */
                        /*      1000          ,     2000    */
                        /*          lastname           c_last        */
                        if (c_id <= 1000) {
                              lastname(c_id - 1, c_last);
                        } else {
                              lastname(nurand(255, 0, 999,C_LOAD), c_last);
                        }

                        /*    ,                    */
                        make_address(c_street_1, c_street_2, c_city, c_state, c_zip);

                        make_number_string(16, 16, c_phone);

                        /* c_credit          10% BC, 90%  GC         */
                        if (random_number(0, 9)) {
                              strcpy(c_credit, "GC");
                        } else {
                              strcpy(c_credit, "BC");
                        }

                        /* c_discount          */
        /* c_discount /= d_10000;*/
                        c_discount = random_number(0, 5000);

                        /* c_data         */
                        make_alpha_string(300, 500, c_data);

                        /* record   :    : 1997-02-27 */
                  memset(customer_cp , ' ' , CUSTOMER_SIZE)        ;
            len_i = strlen(c_first)                    ;
```

```c
        strncpy(customer_cp->c_first  ,c_first  ,len_i) ;
        strncpy(customer_cp->c_middle ,c_middle ,2    ) ;
        len_i = strlen(c_last)                      ;
        strncpy(customer_cp->c_last   ,c_last   ,len_i) ;

        len_i = strlen(c_street_1)                  ;
        strncpy(customer_cp->c_street_1,c_street_1,len_i) ;
        len_i = strlen(c_street_2)                  ;
        strncpy(customer_cp->c_street_2,c_street_2,len_i) ;
        len_i = strlen(c_city)                      ;
        strncpy(customer_cp->c_city   ,c_city   ,len_i) ;
        strncpy(customer_cp->c_state  ,c_state  ,2    ) ;
        strncpy(customer_cp->c_zip    ,c_zip    ,9    ) ;
        strncpy(customer_cp->c_phone  ,c_phone  ,16   ) ;
        strncpy(customer_cp->c_since  ,c_since  ,14   ) ;
        strncpy(customer_cp->c_credit ,c_credit ,2    ) ;

        len_i = strlen(c_data)                      ;
        strncpy(customer_cp->c_data   ,c_data   ,len_i) ;

                /* record   :    : 96/09/09 fukui */
        /*
                c_id_1 = c_id / 256;
                c_id_2 = c_id % 256;
        */
        /* 97-02-18 c_id         short -> int */
                customer_cp->c_id_1 = c_id / T256;
                customer_cp->c_id_2 = (c_id - (c_id_1 * T256))
/ D256;
                customer_cp->c_id_3 = (c_id - (c_id_1 * T256)
                        - (c_id_2 * D256)) / 256;
                customer_cp->c_id_4 = c_id % T256;

                customer_cp->c_d_id_1 = c_d_id / 256;
                customer_cp->c_d_id_2 = c_d_id % 256;
                customer_cp->c_w_id_1 = c_w_id / 256;
                customer_cp->c_w_id_2 = c_w_id % 256;
                customer_cp->c_credit_lim_1 = 0x00;/*
c_credit_lim:+50000.00*/
                customer_cp->c_credit_lim_2 = 0x00;
                customer_cp->c_credit_lim_3 = 0x00;
                customer_cp->c_credit_lim_4 = 0x50;
                customer_cp->c_credit_lim_5 = 0x00;
                customer_cp->c_credit_lim_6 = 0x00;
                customer_cp->c_credit_lim_7 = 0x0c;
                customer_cp->c_discount_1 = c_discount / 256;
                customer_cp->c_discount_2 = c_discount %
256;
                customer_cp->c_balance_1 = 0x00;/* c_balance:
-10.00 */
                customer_cp->c_balance_2 = 0x00;
                customer_cp->c_balance_3 = 0x00;
                customer_cp->c_balance_4 = 0x00;
                customer_cp->c_balance_5 = 0x01;
                customer_cp->c_balance_6 = 0x00;
                customer_cp->c_balance_7 = 0x0d;
                customer_cp->c_ytd_payment_1 = 0x00; /*
c_ytd_payment:+10.00*/
                customer_cp->c_ytd_payment_2 = 0x00;
                customer_cp->c_ytd_payment_3 = 0x00;
                customer_cp->c_ytd_payment_4 = 0x00;
                customer_cp->c_ytd_payment_5 = 0x01;
                customer_cp->c_ytd_payment_6 = 0x00;

                customer_cp->c_ytd_payment_7 = 0x0c;
                customer_cp->c_payment_cnt_1 = c_payment_cnt / 256;
                customer_cp->c_payment_cnt_2 = c_payment_cnt % 256;
                customer_cp->c_delivery_cnt_1 = c_delivery_cnt / 256;
                customer_cp->c_delivery_cnt_2 = c_delivery_cnt % 256;

        customer_cp    = customer_cp   + 1 ;
        customer_lpcnt = customer_lpcnt + 1 ;

        if ( customer_lpcnt == CUSTOMER_COUNT )
          {
           write(op_customer ,
                 customer_ap ,
                 (size_t)CUSTOMER_SIZE * (size_t)CUSTOMER_COUNT) ;
           customer_cp   = (customer_str *)customer_ap ;
           customer_lpcnt = 0          ;
          }

            }
        }
        filecount++;
    }

    /* 1997-02-18  fprintf -> sprintf + fwrite   */
    if ( customer_lpcnt != 0 )
      {
       write(op_customer ,
             customer_ap ,
             (size_t)CUSTOMER_SIZE * (size_t)customer_lpcnt) ;
      }

    /*              */
    close(op_customer) ;

  /*              */
  free(customer_ap) ;

    /*              */
    return;

}
/*
 *    Function: history()
 *    Description: history            .
 *    Parameters: 1. base_ware,
 *    Parameters: 2. last_ware,
 *    Grobals Ref: yyyymmddhhmmss,
 *    Grobals Out: nothing
 *    Returns    : nothing
 */

void
history(base_ware,last_ware)
int    base_ware;
int    last_ware;
{
        /*       */
        int            filecount = 1;
        int            outfilecount;
        char        filename2[64];
        /* 1997-02-18  fprintf -> sprintf + fwrite     */
    int        history_lpcnt           ; /*              */
    char        *history_ap            ; /*              */
```

```c
    history_str *history_cp          ; /*              */

        h_amount = 10;
        strcpy(h_date, yyyymmddhhmmss);

        outfilecount = ((base_ware-1)/10) +1;
        sprintf(filename2 , "%s/HI%d_%d" ,fileout, base_ware,
last_ware);
        /*if ((fst6 = fopen(filename2 , "w"))==NULL){   1997.02.24 */
        if ((op_history = open(
filename2,O_WRONLY|O_CREAT|O_TRUNC,
            S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH
))==NULL){
            printf("wttpcc:    : %s:            \n" , filename2);
            exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite     */
      /* HISTORY                      */
      history_ap = (char
*)malloc((size_t)(HISTORY_SIZE*HISTORY_COUNT)  );
      if ( history_ap == NULL )                /*          */
       {                        /*          */
        printf("Malloc failed.(history)\n")    ; /*          */
        exit(1)                    ; /*          */
       }                        /*          */
      history_cp    = (history_str *)history_ap    ; /*          */
      history_lpcnt  = 0              ; /*          */

        /* w_id          count_ware        */
        for (h_c_w_id = base_ware; h_c_w_id <= last_ware;
h_c_w_id++){

            fprintf(stderr,"wttpcc:    : HISTORY %d/%d  %d      \n"
                        ,h_c_w_id, base_ware,last_ware);

            /* d_id            DIST_PER_WARE          */
            for (h_c_d_id = 1; h_c_d_id <= DIST_PER_WARE;
h_c_d_id++){

                /* c_id              CUST_PER_DIST          */
                /* coustomer, history              */
                for (h_c_id = 1; h_c_id <= CUST_PER_DIST;
h_c_id++) {

                    /*h_c_w_id = c_w_id;*/
                    /*h_c_d_id = c_d_id;*/
                    /*h_c_id = c_id;   */
                    h_w_id = h_c_w_id;
                    h_d_id = h_c_d_id;

                /* h_data        */
                    make_alpha_string(12, 24, h_data);

                    /* record    :    : 1997-02-27 */
                  memset(history_cp->h_data , ' ' , 24)    ;
              len_i = strlen(h_data)                ;
              strncpy(history_cp->h_data,h_data,len_i) ;
              strncpy(history_cp->h_date,h_date,14   ) ;

                /* 97-02-18 h_c_id        short -> int */
                    history_cp->h_c_id_1= h_c_id / T256;
```

```c
                history_cp->h_c_id_2=(h_c_id - (h_c_id_1 * T256)) / D256;
                history_cp->h_c_id_3=(h_c_id - (h_c_id_1 * T256)
                        - (h_c_id_2 * D256)) /256;
                history_cp->h_c_id_4= h_c_id % T256;

                history_cp->h_c_d_id_1 = h_c_d_id / 256;
                history_cp->h_c_d_id_2 = h_c_d_id % 256;
                history_cp->h_c_w_id_1 = h_c_w_id / 256;
                history_cp->h_c_w_id_2 = h_c_w_id % 256;
                history_cp->h_d_id_1 = h_d_id / 256;
                history_cp->h_d_id_2 = h_d_id % 256;
                history_cp->h_w_id_1 = h_w_id / 256;
                history_cp->h_w_id_2 = h_w_id % 256;
                history_cp->h_amount_1 = h_amount / T256;
                history_cp->h_amount_2 = (h_amount - (h_amount_1*T256))
                                            / D256;

                history_cp->h_amount_3 = (h_amount - (h_amount_1*T256)
                        - (h_amount_2*D256)) / D256;
                history_cp->h_amount_4 = h_amount % T256;

        history_cp   = history_cp   + 1 ;
        history_lpcnt = history_lpcnt + 1 ;

        if ( history_lpcnt == HISTORY_COUNT )
          {
           write(op_history ,
                history_ap ,
                (size_t)HISTORY_SIZE * (size_t)HISTORY_COUNT) ;
            history_cp   = (history_str *)history_ap  ;
            history_lpcnt = 0          ;
          }


            }
        }
        filecount++;
    }

    /* 1997-02-18  fprintf -> sprintf + fwrite   */
    if ( history_lpcnt != 0 )
     {
      write(op_history ,
            history_ap ,
            (size_t)HISTORY_SIZE  * (size_t)history_lpcnt) ;
     }

    /*              */
    close(op_history ) ;

  /*              */
  free(history_ap) ;

    /*              */
    return;

}

/*
 *    Function: orders()
 *    Description: orders, order_line, new_order          .
 *    Parameters: 1. base_ware,
 *    Parameters: 1. last_ware,
 *    Grobals Ref: yyyymmddhhmmss,
 *    Grobals Out: nothing
```

```
*       Returns     : nothing
 */

void
orders(base_ware,last_ware)
int    base_ware;
int    last_ware;
{
        /*      */
        double      d_100 = 100;
        int              filecount = 1;
        int              outfilecount;
        char         filename1[64];
        char         filename2[64];
        char         filename3[64];
        short        d_id;
        short        w_id;
        int              o_id;
        /* 1997-02-18  fprintf -> sprintf + fwrite     */
        int          orders_lpcnt             ; /*                */
        char        *orders_ap               ; /*                */
        orders_str   *orders_cp              ; /*                 */
        int          orderline_lpcnt         ; /*                */
        char        *orderline_ap            ; /*                */
        orderline_str *orderline_cp              ; /*                */

        o_all_local = 1;
        ol_quantity = 5;
        outfilecount = ((base_ware-1)/10) + 1;

        /*               */
        sprintf(filename1 , "%s/OS%d_%d" ,fileout, base_ware,
last_ware);
        /*if ((fst7 = fopen(filename1 , "w"))==NULL){   1997.02.24 */
        if ((op_orders = open(
filename1,O_WRONLY|O_CREAT|O_TRUNC,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH
))==NULL){
                printf("wttpcc:       : %s:             \n" , filename1);
                exit(1);
        }
        sprintf(filename2 , "%s/NO%d_%d" ,fileout, base_ware,
last_ware);
        if ((fst8 = fopen(filename2 , "w"))==NULL){
                printf("wttpcc:       : %s:             \n" , filename2);
                exit(1);
        }
        sprintf(filename3 , "%s/OL%d_%d" ,fileout, base_ware,
last_ware);
        /*if ((fst9 = fopen(filename3 , "w"))==NULL){   1997.02.24 */
        if ((op_orderline = open(
filename3,O_WRONLY|O_CREAT|O_TRUNC,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH
))==NULL){
                printf("wttpcc:       : %s:             \n" , filename3);
                exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite     */
    /* ORDERS                          */
    /* ORDERLINE                             */
    orders_ap = (char
*)malloc((size_t)(ORDERS_SIZE*ORDERS_COUNT)
```

```
                                + (ORDERLINE_SIZE*ORDERLINE_COUNT));
        if ( orders_ap == NULL )                   /*                */
         {                                          /*                */
           printf("Malloc failed.(orders)\n")       ;/*                */
           exit(1)                                  ;/*                */
         }                                          /*                */
        orderline_ap    = orders_ap + (ORDERS_SIZE*ORDERS_COUNT) ;
        orders_cp      = (orders_str *)orders_ap      ;/*                */
        orderline_cp    = (orderline_str *)orderline_ap;/*                */
        orders_lpcnt    = 0                        ;/*                */
        orderline_lpcnt = 0                        ;/*                */

        /* w_id            , count_ware        */
        for (w_id = base_ware; w_id <= last_ware; w_id++){

                fprintf(stderr,"wttpcc:   : ORDERS/O.LINE/N.ORDER %d/%d  %d
\n"
                                ,w_id, base_ware, last_ware);

            /* d_id            , DIST_PER_WARE        */
            for (d_id = 1; d_id <= DIST_PER_WARE; d_id++) {

                /* o_c_id                         */
                init_permutation();

            /* o_id 1-ORD_PER_DIST          , ORD_PER_DIST    */
            /* orders                     */
                for (o_id = 1; o_id <= ORD_PER_DIST; o_id++) {

                        no_o_id = o_id;
                        no_w_id = w_id;
                        no_d_id = d_id;
                        o_id = o_id;
                        o_w_id = w_id;
                        o_d_id = d_id;
                        o_ol_cnt = random_number(5, 15);
                        strcpy(o_entry_d, yyyymmddhhmmss);
                        ol_o_id = o_id;
                        ol_w_id = w_id;
                        ol_d_id = d_id;
                        ol_supply_w_id = w_id;

                        /* o_c_id                        1997-02-18 */
            /* o_c_id         init_permutation              */
            /*      1 CUST_PER_DIST                  . */
                /*                               */
            counter++;
            /* ocid  counter-1                      */
                o_c_id = ocid[counter - 1] ;

                        orders_cp->o_entry_d_v1 = NNUL_V;
                        orders_cp->o_entry_d_v2 = NNUL_V;
        strncpy(orders_cp->o_entry_d,o_entry_d,14) ;

                        /* o_id>2100   ,        */
                        /*           ,           new_order */
                        if (o_id > (ORD_PER_DIST - NEWORDS)){
                            /* ocarrier_id        */
                            /* o_carrier_id = 0x00; *//* NULL   */
                                orders_cp->o_carrier_id_v1 = NUL_V;
                                orders_cp->o_carrier_id_v2 = NUL_V;
                                orders_cp->o_carrier_id_1 = 0x00;
                                orders_cp->o_carrier_id_2 = 0x00;
```

```
                        /* orders record  :    : 96/09/09 fukui                      } else {
*/                                                                                        /* ocarrier_id        */
                        /* ( NULL  ) */                                                   o_carrier_id = random_number(1, 10);
                        orders_cp->o_id_v1 = NNUL_V;
                        orders_cp->o_id_v2 = NNUL_V;                                      orders_cp->o_carrier_id_v1 = NNUL_V;
                        orders_cp->o_id_1 = o_id / T256;                                  orders_cp->o_carrier_id_v2 = NNUL_V;
                        orders_cp->o_id_2 = (o_id-                                        orders_cp->o_carrier_id_1 = o_carrier_id / 256;
(o_id_1*T256))/D256;                                                                     orders_cp->o_carrier_id_2 = o_carrier_id % 256;
                        orders_cp->o_id_3 = (o_id-                                        /* order record  :    : 96/09/09 fukui */
(o_id_1*T256)-(o_id_2*D256))/256;                                                        orders_cp->o_id_v1 = NNUL_V;
                        orders_cp->o_id_4 = o_id % T256;                                  orders_cp->o_id_v2 = NNUL_V;
                                                                                         orders_cp->o_id_1 = o_id / T256;
                        orders_cp->o_d_id_v1 = NNUL_V;                                    orders_cp->o_id_2 = (o_id-(o_id_1*T256))/D256;
                        orders_cp->o_d_id_v2 = NNUL_V;                                    orders_cp->o_id_3 = (o_id-(o_id_1*T256)-
                        orders_cp->o_d_id_1 = o_d_id / 256;         (o_id_2*D256))/256;
                        orders_cp->o_d_id_2 = o_d_id % 256;                               orders_cp->o_id_4 = o_id % T256;

                        orders_cp->o_w_id_v1 = NNUL_V;                                    orders_cp->o_d_id_v1 = NNUL_V;
                        orders_cp->o_w_id_v2 = NNUL_V;                                    orders_cp->o_d_id_v2 = NNUL_V;
                        orders_cp->o_w_id_1 = o_w_id / 256;                               orders_cp->o_d_id_1 = o_d_id / 256;
                        orders_cp->o_w_id_2 = o_w_id % 256;                               orders_cp->o_d_id_2 = o_d_id % 256;

          /* 97-02-18 o_c_id      short->int */                                          orders_cp->o_w_id_v1 = NNUL_V;
                        orders_cp->o_c_id_v1 = NNUL_V;                                    orders_cp->o_w_id_v2 = NNUL_V;
                        orders_cp->o_c_id_v2 = NNUL_V;                                    orders_cp->o_w_id_1 = o_w_id / 256;
                        orders_cp->o_c_id_1 = o_c_id / T256;                              orders_cp->o_w_id_2 = o_w_id % 256;
                        orders_cp->o_c_id_2 = (o_c_id-                      /* 97-02-18 o_c_id      short->int */
(o_c_id_1*T256)) / D256;                                                                 orders_cp->o_c_id_v1 = NNUL_V;
                        orders_cp->o_c_id_3 = (o_c_id-                                    orders_cp->o_c_id_v2 = NNUL_V;
(o_c_id_1*T256)                                                                          orders_cp->o_c_id_1 = o_c_id / T256;
                                                                                         orders_cp->o_c_id_2 = (o_c_id-(o_c_id_1*T256)) /
                            -(o_c_id_2*D256))/256;                         D256;
                        orders_cp->o_c_id_4 = o_c_id % T256;                              orders_cp->o_c_id_3 = (o_c_id-(o_c_id_1*T256)
                                                                                             -(o_c_id_2*D256))/256;
                        orders_cp->o_ol_cnt_v1 = NNUL_V;                                  orders_cp->o_c_id_4 = o_c_id % T256;
                        orders_cp->o_ol_cnt_v2 = NNUL_V;
                        orders_cp->o_ol_cnt_1 = o_ol_cnt / 256;                           orders_cp->o_ol_cnt_v1 = NNUL_V;
                        orders_cp->o_ol_cnt_2 = o_ol_cnt % 256;                           orders_cp->o_ol_cnt_v2 = NNUL_V;
                                                                                         orders_cp->o_ol_cnt_1 = o_ol_cnt / 256;
                        orders_cp->o_all_local_v1 = NNUL_V;                               orders_cp->o_ol_cnt_2 = o_ol_cnt % 256;
                        orders_cp->o_all_local_v2 = NNUL_V;
                        orders_cp->o_all_local_1 = o_all_local /                          orders_cp->o_all_local_v1 = NNUL_V;
256;                                                                                      orders_cp->o_all_local_v2 = NNUL_V;
                        orders_cp->o_all_local_2 = o_all_local %                          orders_cp->o_all_local_1 = o_all_local / 256;
256;                                                                                      orders_cp->o_all_local_2 = o_all_local % 256;

                        /* Neworder record  :    : 96/09/09                           }
fukui */
                        no_o_id_1 = no_o_id / T256;                      orders_cp  = orders_cp  + 1 ;
                        no_o_id_2 = (no_o_id-                            orders_lpcnt = orders_lpcnt + 1 ;
(no_o_id_1*T256))/D256;                                                 if ( orders_lpcnt == ORDERS_COUNT )
                        no_o_id_3 =(no_o_id-(no_o_id_1*T256)-            {
(no_o_id_2*D256))/256;                                                   write(op_orders ,
                        no_o_id_4 = no_o_id % T256;                            orders_ap ,
                        no_d_id_1 = no_d_id / 256;                               (size_t)ORDERS_SIZE * (size_t)ORDERS_COUNT) ;
                        no_d_id_2 = no_d_id % 256;                          orders_cp  = (orders_str *)orders_ap  ;
                        no_w_id_1 = no_w_id / 256;                          orders_lpcnt = 0        ;
                        no_w_id_2 = no_w_id % 256;                       }
                        fprintf(fst8 ,"%c%c%c%c"
                   "%c%c%c%c",                                                            /* ol_number         ,o_olcnt   */
                 no_o_id_1,no_o_id_2,no_o_id_3,no_o_id_4,                                 /* order_line                    */
                 no_d_id_1,no_d_id_2,no_w_id_1,no_w_id_2);
```

```
                for (ol_number = 1; ol_number <= o_ol_cnt;
ol_number++)
            {

                /* ol_i_id, ol_dist_info       */
                /* ol_i_id = random_number(1,
MAXITEMS); */

        /*    :961127:K.Fukui: I_ID      (main   ) */

                ol_i_id = random_number( 1,
MAXITEMS / I_ID_Rand_by );
                ol_i_id = ol_i_id * I_ID_Rand_by;

                make_alpha_string(24, 24, ol_dist_info);
                orderline_cp->ol_dist_info_v1 =
NNUL_V;
                orderline_cp->ol_dist_info_v2 =
NNUL_V;
        strncpy(orderline_cp->ol_dist_info,ol_dist_info,24) ;

                /* o_id>2100     */
                /* ol_amount      */
                /*          ,       */
                if (o_id > (CUST_PER_DIST -
NEWORDS))
            {
                /*ol_amount /= d_100;  */
                ol_amount = random_number(1,
999999);

                /* orderline         */

            /*ol_delivery_d = NULL;*/
                orderline_cp->ol_delivery_d_v1 =
NUL_V;
                orderline_cp->ol_delivery_d_v2 =
NUL_V;

                orderline_cp->ol_o_id_v1 =
NNUL_V;
                orderline_cp->ol_o_id_v2 =
NNUL_V;
                orderline_cp->ol_o_id_1 = ol_o_id /
T256;
                orderline_cp->ol_o_id_2 = (ol_o_id-
(ol_o_id_1*T256))
                            /D256;
                orderline_cp->ol_o_id_3 = (ol_o_id-
(ol_o_id_1*T256)
                    -(ol_o_id_2*D256))/256;
                orderline_cp->ol_o_id_4 = ol_o_id
% T256;

                orderline_cp->ol_d_id_v1 =
NNUL_V;
                orderline_cp->ol_d_id_v2 =
NNUL_V;
                orderline_cp->ol_d_id_1 = ol_d_id /
256;
                orderline_cp->ol_d_id_2 = ol_d_id
% 256;
```

```
                            (ol_i_id_1*T256))
                            /D256;

                            (ol_i_id_1*T256)

                            ol_supply_w_id / 256;

                            ol_supply_w_id % 256;

256;

                            256;

                            T256;

        } else {
            ol_amount = 0;

            strcpy(ol_delivery_d, yyyymmddhhmmss);
            orderline_cp->ol_delivery_d_v1 = NNUL_V;
            orderline_cp->ol_delivery_d_v2 = NNUL_V;
        strncpy(orderline_cp->ol_delivery_d,
            ol_delivery_d,14);
```

```
            orderline_cp->ol_w_id_v1 = NNUL_V;
            orderline_cp->ol_w_id_v2 = NNUL_V;
            orderline_cp->ol_w_id_1 = ol_w_id / 256;
            orderline_cp->ol_w_id_2 = ol_w_id % 256;

            orderline_cp->ol_number_v1 = NNUL_V;
            orderline_cp->ol_number_v2 = NNUL_V;
            orderline_cp->ol_number_1 = ol_number / 256;
            orderline_cp->ol_number_2 = ol_number % 256;

            orderline_cp->ol_i_id_v1 = NNUL_V;
            orderline_cp->ol_i_id_v2 = NNUL_V;
            orderline_cp->ol_i_id_1 = ol_i_id / T256;
            orderline_cp->ol_i_id_2 = (ol_i_id-

                            /D256;
            orderline_cp->ol_i_id_3 = (ol_i_id-

                    -(ol_i_id_2*D256))/256;
            orderline_cp->ol_i_id_4 = ol_i_id % T256;

            orderline_cp->ol_supply_w_id_v1 = NNUL_V;
            orderline_cp->ol_supply_w_id_v2 = NNUL_V;
            orderline_cp->ol_supply_w_id_1 =

            orderline_cp->ol_supply_w_id_2 =

            orderline_cp->ol_quantity_v1 = NNUL_V;
            orderline_cp->ol_quantity_v2 = NNUL_V;
            orderline_cp->ol_quantity_1 = ol_quantity / 256;
            orderline_cp->ol_quantity_2 = ol_quantity %

            orderline_cp->ol_amount_v1 = NNUL_V;
            orderline_cp->ol_amount_v2 = NNUL_V;
            orderline_cp->ol_amount_1 = ol_amount / T256;
            orderline_cp->ol_amount_2 = (ol_amount
                -(ol_amount_1*T256))/D256;
            orderline_cp->ol_amount_3 = (ol_amount
                -(ol_amount_1*T256)
                -(ol_amount_2*D256))/256;
            orderline_cp->ol_amount_4 = ol_amount %

            orderline_cp->ol_o_id_v1 = NNUL_V;
            orderline_cp->ol_o_id_v2 = NNUL_V;
            orderline_cp->ol_o_id_1 = ol_o_id / T256;
            orderline_cp->ol_o_id_2 = (ol_o_id
              -(ol_o_id_1*T256))/D256;
            orderline_cp->ol_o_id_3 = (ol_o_id
              -(ol_o_id_1*T256)
              -(ol_o_id_2*D256))/256;
            orderline_cp->ol_o_id_4 = ol_o_id % T256;
```

```
                        orderline_cp->ol_d_id_v1 =
NNUL_V;
                        orderline_cp->ol_d_id_v2 =
NNUL_V;
                        orderline_cp->ol_d_id_1 = ol_d_id
/ 256;
                        orderline_cp->ol_d_id_2 = ol_d_id
% 256;

                        orderline_cp->ol_w_id_v1 =
NNUL_V;
                        orderline_cp->ol_w_id_v2 =
NNUL_V;
                        orderline_cp->ol_w_id_1 = ol_w_id
/ 256;
                        orderline_cp->ol_w_id_2 = ol_w_id
% 256;

                        orderline_cp->ol_number_v1 =
NNUL_V;
                        orderline_cp->ol_number_v2 =
NNUL_V;
                        orderline_cp->ol_number_1 =
ol_number / 256;
                        orderline_cp->ol_number_2 =
ol_number % 256;

                        orderline_cp->ol_i_id_v1 =
NNUL_V;
                        orderline_cp->ol_i_id_v2 =
NNUL_V;
                        orderline_cp->ol_i_id_1 = ol_i_id /
T256;
                        orderline_cp->ol_i_id_2 = (ol_i_id
                          -(ol_i_id_1*T256))/D256;
                        orderline_cp->ol_i_id_3 = (ol_i_id
                          -(ol_i_id_1*T256)
                          -(ol_i_id_2*D256))/256;
                        orderline_cp->ol_i_id_4 = ol_i_id %
T256;

                        orderline_cp->ol_supply_w_id_v1
= NNUL_V;
                        orderline_cp->ol_supply_w_id_v2
= NNUL_V;
                        orderline_cp->ol_supply_w_id_1 =
ol_supply_w_id / 256;
                        orderline_cp->ol_supply_w_id_2 =
ol_supply_w_id % 256;

                        orderline_cp->ol_quantity_v1 =
NNUL_V;
                        orderline_cp->ol_quantity_v2 =
NNUL_V;
                        orderline_cp->ol_quantity_1 =
ol_quantity / 256;
                        orderline_cp->ol_quantity_2 =
ol_quantity % 256;

                        orderline_cp->ol_amount_v1 =
NNUL_V;
```

```
                        orderline_cp->ol_amount_v2 = NNUL_V;
                        orderline_cp->ol_amount_1 = ol_amount / T256;
                        orderline_cp->ol_amount_2 = (ol_amount
                          -(ol_amount_1*T256))/D256;
                        orderline_cp->ol_amount_3 = (ol_amount
                          -(ol_amount_1*T256)
                          -(ol_amount_2*D256))/256;
                        orderline_cp->ol_amount_4 = ol_amount % T256;

                                        }

      orderline_cp    = orderline_cp    + 1 ;
      orderline_lpcnt = orderline_lpcnt + 1 ;
      if ( orderline_lpcnt == ORDERLINE_COUNT )
        {
          write(op_orderline ,
            orderline_ap ,
            (size_t)ORDERLINE_SIZE * (size_t)ORDERLINE_COUNT);
          orderline_cp    = (orderline_str *)orderline_ap ;
          orderline_lpcnt = 0           ;
        }
                }
            }
        }
        filecount++;
}

if ( orders_lpcnt != 0 )
  {
    write(op_orders ,
        orders_ap ,
        (size_t)ORDERS_SIZE * (size_t)orders_lpcnt) ;
  }
if ( orderline_lpcnt != 0 )
  {
    write(op_orderline ,
        orderline_ap ,
          (size_t)ORDERLINE_SIZE * (size_t)orderline_lpcnt) ;
  }

/*              */
fclose(fst8);
close(op_orders)    ;
close(op_orderline) ;

/*              */
free(orders_ap)    ;

/*              */
return;

}
/*
*      Function: make_address()
*      Description:              .
*      Parameters: 1. str1,    1(        21   )
*                  2. str2,    2(        21   )
*                  3. city,     (        21   )
*                  4. state,    (        3   )
*                  5. zip,      (        10   )
*      Grobals Ref: nothing
*      Grobals Out: nothing
```

```
 *      Returns    : nothing
 */

void
make_address(str1, str2, city, state, zip)
char *str1;
char *str2;
char *city;
char *state;
char *zip;
{
        /* street1      , 10-20                   */
        make_alpha_string(10, 20, str1);

        /* street2      , 10-20                   */
        make_alpha_string(10, 20, str2);

        /* city      , 10-20                   */
        make_alpha_string(10, 20, city);

        /* state      , 2              */
        make_alpha_string(2, 2, state);

        /* zip      , 9              */
        make_number_string(9, 9, zip);

        return;
}

/*
 *      Function: lastname()
 *      Description: lastname          .
 *      Parameters: 1. num, 000-999
 *                  2. name,
 *      Grobals Ref: nothing
 *      Grobals Out: nothing
 *      Returns    : nothing
 */

void
lastname(num, name)
int   num;
char *name;
{
        /*   syllable  c_last                10       */
        static char *syllable[] = {
                "BAR", "OUGHT", "ABLE", "PRI", "PRES",
                "ESE", "ANTI", "CALLY", "ATION", "EING"
                };

        /* syllable[      100     ] name          */
        strcpy(name, syllable[num / 100]);

        /* syllable[      10     ] name          */
        strcat(name, syllable[(num /10) % 10]);

        /* syllable[      1     ] name          */
        strcat(name, syllable[num % 10]);

        return;
}

/*
```

```
 *      Function: make_alpha_string()
 *      Description:                  .
 *      Parameters: 1. num1,
 *                  2. num2,
 *                  3. str,
 *      Grobals Ref: nothing
 *      Grobals Out: nothing
 *      Returns    : int,
 */

int
make_alpha_string(num1, num2, str)
int   num1;
int   num2;
char *str;
{
        int      len;
        int      i;
        short rnum;

        /*                  num1-num2            */
    if (num1 == num2) {
            len = num1;
        } else {
            len = random_number(num1, num2);
        }
        /*                           */
        for ( i = 0; i < len; i++) {
#ifdef rand_str
            /* 0-61              */
            rnum = random_number(0, 61);

            /* 0-25       ,              'a'==x61 (0:a, 1:b, .. , 25:z) */
            if ((0 <= rnum) && (rnum <= 25)) {
                    str[i] = 'a' + rnum;

            /* 26-51       ,         'A'==x41 (26:A, 27:B, .. , 51:Z) */
            } else if ((26 <= rnum) && (rnum <= 51)) {
                    str[i] = 'A' + rnum - 26;

            /* 52-61       ,         '0'==x30 (52:0, 53:1, .. , 61:9) */
            } else if ((52 <= rnum) && (rnum <= 61)) {
                    str[i] = '0' + rnum - 52;
            }
#else
            /* 0-51              */
        rnum = rand()%52 ;          /* 1997.03.11   */

            /* 0-25       ,              'a'==x61 (0:a, 1:b, .. , 25:z) */
            if ((0 <= rnum) && (rnum <= 25)) {
                    str[i] = 'a' + rnum;

            /* 26-51       ,         'A'==x41 (26:A, 27:B, .. , 51:Z) */
            } else if ((26 <= rnum) && (rnum <= 51)) {
                    str[i] = 'A' + rnum - 26;

            }
#endif
        }
        /*                  */
/*    if (num1 != num2) {*/
            str[len] = '\0';
/*    }*/
```

```
        /*                          */
        return(len);
}

/*
 *    Function: make_number_string()
 *    Description:              .
 *    Parameters: 1. num1,
 *                2. num2,
 *                3. str,
 *    Grobals Ref: nothing
 *    Grobals Out: nothing
 *    Returns    : int,
 */

int
make_number_string(num1, num2, str)
int    num1;
int    num2;
char *str;
{
        int        len;
        int        i;
        short rnum;

        /*                    num1-num2          */
        if (num1 == num2) {
              len = num1;
        } else {
              len = random_number(num1, num2);
        }
        /*                          */
        for ( i = 0; i < len; i++) {

#ifdef rand_str
              /* 0-9            */
              rnum = random_number(0, 9);

              /* 0-9              str       */
              str[i] = '0' + rnum;
#else
        str[i] = (char)(rand()%10+'0');

#endif
        }

        /*                */
        str[len] = '\0';

        /*                        */
        return(len);
}

/*
 *    Function: random_number()
 *    Description:            .
 *    Parameters: 1. num1,
 *                2. num2,
 *    Grobals Ref: nothing
 *    Grobals Out: nothing
 *    Returns    : int,
 */
```

```
#ifdef call_rand
/* 1997-02-18 mac        */
int
random_number(num1, num2)
int    num1;
int    num2;
{
        int    value;

        /* num1-num2            */
        value = lrand48() % (num2 - num1 + 1) + num1;

        /*                      */
        return(value);
}
#endif
/*
 *    Function: set_seed()
 *    Description:              .
 *    Parameters: 1. seedval,
 *    Grobals Ref: nothing
 *    Grobals Out: nothing
 *    Returns    : nothing
 */

void
set_seed(seedval)
int    seedval;
{
        /*            */
        srand(seedval) ;                /* 1997-02-18        */
        srand48(seedval);

        return;
}

/*
 *    Function: nurand()
 *    Description:                .
 *    Parameters: 1. a,
 *                2. x,
 *                3. y,
 *    Grobals Ref: nothing
 *    Grobals Out: nothing
 *    Returns    : nothing
 */
/* 1997-02-18 TAB ID 221(c_last NURand    C)        */
int
nurand(a, x, y, c)
int    a;
int    x;
int    y;
int c;
{
        int    value;

        /*                */
        value = (((random_number(0, a) | random_number(x, y)) + c) %
              (y - x + 1)) + x;

        /*                      */
        return(value);
}
```

```
/*
 *      Function: init_permutation()
 *      Description: o_c_id    1  CUST_PER_DIST
 *                      .
 *      Parameters: nothing
 *      Grobals Ref: nothing
 *      Grobals Out: 1. ocid, o_c_id
 *                   2. counter,
 *      Returns    : nothing
 */

void
init_permutation()
{
      short cnt;
      short replace;
      short work;

      /*          ocid  1-CUST_PER_DIST           */
      for (cnt = 0; cnt < CUST_PER_DIST; cnt++){
           ocid[cnt] = cnt + 1;
      }

      /* ocid                     */
      for (cnt = 0; cnt < CUST_PER_DIST; cnt++){
           replace = random_number(1, CUST_PER_DIST);
           work = ocid[cnt];
           ocid[cnt] = ocid[replace - 1];
           ocid[replace -1] = work;
      }

      /*                   */
      counter = 0;
}
```

≡ *B*

Appendix C: Tunable Parameters **C≡**

This Appendix contains the configuration information for the operating system, the RDBMS and Tuxedo.

Operating System Configuration Values

The Solaris 8 kernel configuration parameters set in the file /etc/system are given below.

Solaris 8 Configuration File for Server, Clients, and RTEs

```
server:
set msgsys:msginfo_msgmap = 200
set msgsys:msginfo_msgmax = 16384
set msgsys:msginfo_msgmnb = 32768
set msgsys:msginfo_msgmni = 512
set msgsys:msginfo_msgseg = 31744
set semsys:seminfo_semmni = 24576
set semsys:seminfo_semmns = 18432
set semsys:seminfo_semmnu = 6144
set semsys:seminfo_semmsl = 32
set shmsys:shminfo_shmmax = 0xffffffff
set shmsys:shminfo_shmmni = 1024
set shmsys:shminfo_shmseg = 512
set pt_cnt = 300
forceload: drv/vxio
forceload: drv/vxspec
set vxio:vol_maxio=2048
client(c1):
set pt_cnt=4096
set shmsys:shminfo_shmmax=0xffffffff
set shmsys:shminfo_shmseg=600
```

185

```
set shmsys:shminfo_shmmni=10
set msgsys:msginfo_msgmni=4096
set msgsys:msginfo_msgmax=2048
set msgsys:msginfo_msgmnb=800000
set msgsys:msginfo_msgmap=200000
set msgsys:msginfo_msgseg=10000
set msgsys:msginfo_msgssz=2048
set msgsys:msginfo_msgtql=5000
set semsys:seminfo_semmns=5000
set semsys:seminfo_semmni=5000
set semsys:seminfo_semmsl=5000
set semsys:seminfo_semmap=5000
set semsys:seminfo_semume=1
set semsys:seminfo_semmnu=5000
set autoup = 300
client(c2..c8):
set pt_cnt=4096
set shmsys:shminfo_shmmax=0xffffffff
set shmsys:shminfo_shmseg=600
set shmsys:shminfo_shmmni=10
set msgsys:msginfo_msgmni=4096
set msgsys:msginfo_msgmax=2048
set msgsys:msginfo_msgmnb=10000
set msgsys:msginfo_msgmap=10000
set msgsys:msginfo_msgseg=10000
set msgsys:msginfo_msgssz=2048
set msgsys:msginfo_msgtql=5000
set semsys:seminfo_semmns=5000
set semsys:seminfo_semmni=5000
set semsys:seminfo_semmsl=5000
set semsys:seminfo_semmap=5000
set semsys:seminfo_semume=1
set semsys:seminfo_semmnu=5000
set tune_t_flckrec=2048
set rlim_fd_max=12288
```

RDBMS Configuration values

```
::::::::::::::
RDBSTART
::::::::::::::
#!/usr/bin/sh -xf
#

/misc/bin/fastsu psradm -n 0
/misc/bin/fastsu psradm -n 1
/misc/bin/fastsu psradm -n 2
/misc/bin/fastsu psradm -n 3

/misc/bin/fastsu psradm -i 1
/misc/bin/fastsu psradm -i 2
/misc/bin/fastsu psradm -i 3
psrinfo
```

```
cd /opt/FSUNrdb2b/etc
rm rdbsysconfig
ln -s rdbsysconfig.5.4i rdbsysconfig

ls -l /rdbptc/FSUNrdb2b
ls -l /rdbptc/FSUNrdb2b/sbin/rdb2base2

cd /rdbptc/tpcc80/tpcc/b-onsrc
SEINO_RCP_PRT1=000 ; export SEINO_RCP_PRT1
SEINO_RCP_LIMITER=200; export SEINO_RCP_LIMITER
SEINO_RCP_MSIZE=4000; export SEINO_RCP_MSIZE
#KARI_NOT_W_ST_CU=1; export KARI_NOT_W_ST_CU
RDB_XCM_NOT_TINF=1; export RDB_XCM_NOT_TINF
RDB_BCM_NOT_TINF=1; export RDB_BCM_NOT_TINF
RDB_NOT_RISTRICT_MODE=1; export RDB_NOT_RISTRICT_MODE
SEINO_RCP_SMALL_FIRST=450; export SEINO_RCP_SMALL_FIRST

RDBDB=TPCC
#
RDBBUFCPU=3  #2  #0 #0 #2  #0
export RDBBUFCPU

# RDB_UNBALANCED_CPUBIND='82 99 99 99 99 99 99 99 99' # 4CPU CPU0=1?/96
# RDB_UNBALANCED_CPUBIND='85 99 99 99 99 99 99 99 99' # 4CPU CPU0=2/96
# RDB_UNBALANCED_CPUBIND='88 99 99 99 99 99 99 99 99' # 4CPU CPU0=3/96
# RDB_UNBALANCED_CPUBIND='91 99 99 99 99 99 99 99 99' # 4CPU CPU0=4/96
RDB_UNBALANCED_CPUBIND='1 99 99 99 99 99 99 99 99' # 4CPU
# RDB_UNBALANCED_CPUBIND='99 99 99 99 99 99 99 99 99' # 1CPU
export RDB_UNBALANCED_CPUBIND

RDBFOPENMAX2=30
export RDBFOPENMAX2

RDBBUFPATH=/rdbptc/tpcc80/tpcc/b-onsrc/rdbcwbuf.wk.5.4
export RDBBUFPATH

### mks Fri Sep 24 23:25:57 PDT 1999
### try it
###mks-oza-Fri Sep 24 18:36:54 PDT 1999
RDB_LOCAL_LP_ADDR=0x3000000
export RDB_LOCAL_LP_ADDR
#RDB_LOCAL_LP_SIZE=576
# RDB_LOCAL_LP_SIZE=448 #400 #384 #464  #  =480 is NG  at 991023
RDB_LOCAL_LP_SIZE=464
export RDB_LOCAL_LP_SIZE

#RDB_LOCAL_LP_ADDR2=0x27000000  #991010
RDB_LOCAL_LP_ADDR2=0x20000000 #0x1d000000 #0x1c000000  #0x21000000  #991010.2
export RDB_LOCAL_LP_ADDR2

#RDB_LOCAL_LP_SIZE2=404    #980614
# RDB_LOCAL_LP_SIZE2=400 #368 #352 #336 #352 #400    #991001 mks
RDB_LOCAL_LP_SIZE2=416
export RDB_LOCAL_LP_SIZE2
```

≡ C

```
#RDBINFKB=1
#export RDBINFKB

date; rdbstart

rdbcrbufcw cw_env.comp.5.4

date

csh sh.crbuf.5.4g.mk
sh sh.crbuf

timex rdbcrbf -f crbuf.aa  2>crbf.wk.out1
grep -v qdg02630 crbf.wk.out1

sh conbf.sh.5.4f

date
sar -r 1 1
swap -s


#for ITEM read
RDBMSG=E
export RDBMSG
setenv  SQLRTENV  sql.env;  export  SQLRTENV
tra.wup  0  0  #wupi.sh #ITEM READ

sar -r 1 1
swap -s

crbfadd.sh
date

cwenvchk

echo "RDBSTART env" > res.doc/RDBSTART.env
env >> res.doc/RDBSTART.env
cwenvchk >>res.doc/RDBSTART.env
crbfadd.sh >>res.doc/RDBSTART.env


rdbpid
if [ $? != 0 ]
then
  rm -f /tmp/ipcrm.csh
  echo "ipcrm_s " > /tmp/ipcrm.csh
  ipcs | grep dba | awk '{ printf "-%s %s ",$1,$2 }' >> /tmp/ipcrm.csh
  echo "\n"  >> /tmp/ipcrm.csh
  chmod a+x /tmp/ipcrm.csh
  /tmp/ipcrm.csh
else
#  rdblog -V -a | nawk 'BEGIN{c=0} $2=="full" {c++} END{for (i=1;i<=c;i++) system("rdblog -R -a")}'
  rdblog -V -a
fi
```

:::::::::::::
conbf_wk2
:::::::::::::
WORK FILE for rdbconbf
TPCC.WAREHOUSE_1_DSI W_1
TPCC.WAREHOUSE_2_DSI W_1
TPCC.WAREHOUSE_3_DSI W_2
TPCC.WAREHOUSE_4_DSI W_2
TPCC.WAREHOUSE_5_DSI W_3
TPCC.WAREHOUSE_6_DSI W_3
TPCC.WAREHOUSE_7_DSI W_4
TPCC.WAREHOUSE_8_DSI W_4
TPCC.WAREHOUSE_9_DSI W_5
TPCC.WAREHOUSE_10_DSI W_5
TPCC.WAREHOUSE_11_DSI W_6
TPCC.WAREHOUSE_12_DSI W_6
TPCC.WAREHOUSE_13_DSI W_1
TPCC.WAREHOUSE_14_DSI W_1
TPCC.WAREHOUSE_15_DSI W_2
TPCC.WAREHOUSE_16_DSI W_2
TPCC.WAREHOUSE_17_DSI W_3
TPCC.WAREHOUSE_18_DSI W_3
TPCC.WAREHOUSE_19_DSI W_4
TPCC.WAREHOUSE_20_DSI W_4
TPCC.WAREHOUSE_21_DSI W_5
TPCC.WAREHOUSE_22_DSI W_5
TPCC.WAREHOUSE_23_DSI W_6
TPCC.WAREHOUSE_24_DSI W_6
TPCC.DISTRICT_1_DSI D_1
TPCC.DISTRICT_2_DSI D_1
TPCC.DISTRICT_3_DSI D_2
TPCC.DISTRICT_4_DSI D_2
TPCC.DISTRICT_5_DSI D_3
TPCC.DISTRICT_6_DSI D_3
TPCC.DISTRICT_7_DSI D_4
TPCC.DISTRICT_8_DSI D_4
TPCC.DISTRICT_9_DSI D_5
TPCC.DISTRICT_10_DSI D_5
TPCC.DISTRICT_11_DSI D_6
TPCC.DISTRICT_12_DSI D_6
TPCC.DISTRICT_13_DSI D_1
TPCC.DISTRICT_14_DSI D_1
TPCC.DISTRICT_15_DSI D_2
TPCC.DISTRICT_16_DSI D_2
TPCC.DISTRICT_17_DSI D_3
TPCC.DISTRICT_18_DSI D_3
TPCC.DISTRICT_19_DSI D_4
TPCC.DISTRICT_20_DSI D_4
TPCC.DISTRICT_21_DSI D_5
TPCC.DISTRICT_22_DSI D_5
TPCC.DISTRICT_23_DSI D_6
TPCC.DISTRICT_24_DSI D_6
TPCC.CUSTOMER_1_DSI C_1
TPCC.CUSTOMER_2_DSI C_1
TPCC.CUSTOMER_3_DSI C_1
TPCC.CUSTOMER_4_DSI C_1

```
TPCC.CUSTOMER_5_DSI C_1
TPCC.CUSTOMER_6_DSI C_1
TPCC.CUSTOMER_7_DSI C_1
TPCC.CUSTOMER_8_DSI C_1
TPCC.CUSTOMER_9_DSI C_2
TPCC.CUSTOMER_10_DSI C_2
TPCC.CUSTOMER_11_DSI C_2
TPCC.CUSTOMER_12_DSI C_2
TPCC.CUSTOMER_13_DSI C_2
TPCC.CUSTOMER_14_DSI C_2
TPCC.CUSTOMER_15_DSI C_2
TPCC.CUSTOMER_16_DSI C_2
TPCC.CUSTOMER_17_DSI C_3
TPCC.CUSTOMER_18_DSI C_3
TPCC.CUSTOMER_19_DSI C_3
TPCC.CUSTOMER_20_DSI C_3
TPCC.CUSTOMER_21_DSI C_3
TPCC.CUSTOMER_22_DSI C_3
TPCC.CUSTOMER_23_DSI C_3
TPCC.CUSTOMER_24_DSI C_3
TPCC.CUSTOMER_25_DSI C_4
TPCC.CUSTOMER_26_DSI C_4
TPCC.CUSTOMER_27_DSI C_4
TPCC.CUSTOMER_28_DSI C_4
TPCC.CUSTOMER_29_DSI C_4
TPCC.CUSTOMER_30_DSI C_4
TPCC.CUSTOMER_31_DSI C_4
TPCC.CUSTOMER_32_DSI C_4
TPCC.CUSTOMER_33_DSI C_5
TPCC.CUSTOMER_34_DSI C_5
TPCC.CUSTOMER_35_DSI C_5
TPCC.CUSTOMER_36_DSI C_5
TPCC.CUSTOMER_37_DSI C_5
TPCC.CUSTOMER_38_DSI C_5
TPCC.CUSTOMER_39_DSI C_5
TPCC.CUSTOMER_40_DSI C_5
TPCC.CUSTOMER_41_DSI C_6
TPCC.CUSTOMER_42_DSI C_6
TPCC.CUSTOMER_43_DSI C_6
TPCC.CUSTOMER_44_DSI C_6
TPCC.CUSTOMER_45_DSI C_6
TPCC.CUSTOMER_46_DSI C_6
TPCC.CUSTOMER_47_DSI C_6
TPCC.CUSTOMER_48_DSI C_6
TPCC.CUSTOMER_49_DSI C_1
TPCC.CUSTOMER_50_DSI C_1
TPCC.CUSTOMER_51_DSI C_1
TPCC.CUSTOMER_52_DSI C_1
TPCC.CUSTOMER_53_DSI C_1
TPCC.CUSTOMER_54_DSI C_1
TPCC.CUSTOMER_55_DSI C_1
TPCC.CUSTOMER_56_DSI C_1
TPCC.CUSTOMER_57_DSI C_2
TPCC.CUSTOMER_58_DSI C_2
TPCC.CUSTOMER_59_DSI C_2
TPCC.CUSTOMER_60_DSI C_2
```

TPCC.CUSTOMER_61_DSI C_2
TPCC.CUSTOMER_62_DSI C_2
TPCC.CUSTOMER_63_DSI C_2
TPCC.CUSTOMER_64_DSI C_2
TPCC.CUSTOMER_65_DSI C_3
TPCC.CUSTOMER_66_DSI C_3
TPCC.CUSTOMER_67_DSI C_3
TPCC.CUSTOMER_68_DSI C_3
TPCC.CUSTOMER_69_DSI C_3
TPCC.CUSTOMER_70_DSI C_3
TPCC.CUSTOMER_71_DSI C_3
TPCC.CUSTOMER_72_DSI C_3
TPCC.CUSTOMER_73_DSI C_4
TPCC.CUSTOMER_74_DSI C_4
TPCC.CUSTOMER_75_DSI C_4
TPCC.CUSTOMER_76_DSI C_4
TPCC.CUSTOMER_77_DSI C_4
TPCC.CUSTOMER_78_DSI C_4
TPCC.CUSTOMER_79_DSI C_4
TPCC.CUSTOMER_80_DSI C_4
TPCC.CUSTOMER_81_DSI C_5
TPCC.CUSTOMER_82_DSI C_5
TPCC.CUSTOMER_83_DSI C_5
TPCC.CUSTOMER_84_DSI C_5
TPCC.CUSTOMER_85_DSI C_5
TPCC.CUSTOMER_86_DSI C_5
TPCC.CUSTOMER_87_DSI C_5
TPCC.CUSTOMER_88_DSI C_5
TPCC.CUSTOMER_89_DSI C_6
TPCC.CUSTOMER_90_DSI C_6
TPCC.CUSTOMER_91_DSI C_6
TPCC.CUSTOMER_92_DSI C_6
TPCC.CUSTOMER_93_DSI C_6
TPCC.CUSTOMER_94_DSI C_6
TPCC.CUSTOMER_95_DSI C_6
TPCC.CUSTOMER_96_DSI C_6
TPCC.ORDERS_1_DSI O_1
TPCC.ORDERS_2_DSI O_1
TPCC.ORDERS_3_DSI O_1
TPCC.ORDERS_4_DSI O_1
TPCC.ORDERS_5_DSI O_1
TPCC.ORDERS_6_DSI O_1
TPCC.ORDERS_7_DSI O_1
TPCC.ORDERS_8_DSI O_1
TPCC.ORDERS_9_DSI O_2
TPCC.ORDERS_10_DSI O_2
TPCC.ORDERS_11_DSI O_2
TPCC.ORDERS_12_DSI O_2
TPCC.ORDERS_13_DSI O_2
TPCC.ORDERS_14_DSI O_2
TPCC.ORDERS_15_DSI O_2
TPCC.ORDERS_16_DSI O_2
TPCC.ORDERS_17_DSI O_3
TPCC.ORDERS_18_DSI O_3
TPCC.ORDERS_19_DSI O_3
TPCC.ORDERS_20_DSI O_3

TPCC.ORDERS_21_DSI O_3
TPCC.ORDERS_22_DSI O_3
TPCC.ORDERS_23_DSI O_3
TPCC.ORDERS_24_DSI O_3
TPCC.ORDERS_25_DSI O_4
TPCC.ORDERS_26_DSI O_4
TPCC.ORDERS_27_DSI O_4
TPCC.ORDERS_28_DSI O_4
TPCC.ORDERS_29_DSI O_4
TPCC.ORDERS_30_DSI O_4
TPCC.ORDERS_31_DSI O_4
TPCC.ORDERS_32_DSI O_4
TPCC.ORDERS_33_DSI O_5
TPCC.ORDERS_34_DSI O_5
TPCC.ORDERS_35_DSI O_5
TPCC.ORDERS_36_DSI O_5
TPCC.ORDERS_37_DSI O_5
TPCC.ORDERS_38_DSI O_5
TPCC.ORDERS_39_DSI O_5
TPCC.ORDERS_40_DSI O_5
TPCC.ORDERS_41_DSI O_6
TPCC.ORDERS_42_DSI O_6
TPCC.ORDERS_43_DSI O_6
TPCC.ORDERS_44_DSI O_6
TPCC.ORDERS_45_DSI O_6
TPCC.ORDERS_46_DSI O_6
TPCC.ORDERS_47_DSI O_6
TPCC.ORDERS_48_DSI O_6
TPCC.ORDERS_49_DSI O_1
TPCC.ORDERS_50_DSI O_1
TPCC.ORDERS_51_DSI O_1
TPCC.ORDERS_52_DSI O_1
TPCC.ORDERS_53_DSI O_1
TPCC.ORDERS_54_DSI O_1
TPCC.ORDERS_55_DSI O_1
TPCC.ORDERS_56_DSI O_1
TPCC.ORDERS_57_DSI O_2
TPCC.ORDERS_58_DSI O_2
TPCC.ORDERS_59_DSI O_2
TPCC.ORDERS_60_DSI O_2
TPCC.ORDERS_61_DSI O_2
TPCC.ORDERS_62_DSI O_2
TPCC.ORDERS_63_DSI O_2
TPCC.ORDERS_64_DSI O_2
TPCC.ORDERS_65_DSI O_3
TPCC.ORDERS_66_DSI O_3
TPCC.ORDERS_67_DSI O_3
TPCC.ORDERS_68_DSI O_3
TPCC.ORDERS_69_DSI O_3
TPCC.ORDERS_70_DSI O_3
TPCC.ORDERS_71_DSI O_3
TPCC.ORDERS_72_DSI O_3
TPCC.ORDERS_73_DSI O_4
TPCC.ORDERS_74_DSI O_4
TPCC.ORDERS_75_DSI O_4
TPCC.ORDERS_76_DSI O_4

TPCC.ORDERS_77_DSI O_4
TPCC.ORDERS_78_DSI O_4
TPCC.ORDERS_79_DSI O_4
TPCC.ORDERS_80_DSI O_4
TPCC.ORDERS_81_DSI O_5
TPCC.ORDERS_82_DSI O_5
TPCC.ORDERS_83_DSI O_5
TPCC.ORDERS_84_DSI O_5
TPCC.ORDERS_85_DSI O_5
TPCC.ORDERS_86_DSI O_5
TPCC.ORDERS_87_DSI O_5
TPCC.ORDERS_88_DSI O_5
TPCC.ORDERS_89_DSI O_6
TPCC.ORDERS_90_DSI O_6
TPCC.ORDERS_91_DSI O_6
TPCC.ORDERS_92_DSI O_6
TPCC.ORDERS_93_DSI O_6
TPCC.ORDERS_94_DSI O_6
TPCC.ORDERS_95_DSI O_6
TPCC.ORDERS_96_DSI O_6
TPCC.ORDERS_IX_1_DSI O_IX_1
TPCC.ORDERS_IX_2_DSI O_IX_1
TPCC.ORDERS_IX_3_DSI O_IX_1
TPCC.ORDERS_IX_4_DSI O_IX_1
TPCC.ORDERS_IX_5_DSI O_IX_1
TPCC.ORDERS_IX_6_DSI O_IX_1
TPCC.ORDERS_IX_7_DSI O_IX_1
TPCC.ORDERS_IX_8_DSI O_IX_1
TPCC.ORDERS_IX_9_DSI O_IX_2
TPCC.ORDERS_IX_10_DSI O_IX_2
TPCC.ORDERS_IX_11_DSI O_IX_2
TPCC.ORDERS_IX_12_DSI O_IX_2
TPCC.ORDERS_IX_13_DSI O_IX_2
TPCC.ORDERS_IX_14_DSI O_IX_2
TPCC.ORDERS_IX_15_DSI O_IX_2
TPCC.ORDERS_IX_16_DSI O_IX_2
TPCC.ORDERS_IX_17_DSI O_IX_3
TPCC.ORDERS_IX_18_DSI O_IX_3
TPCC.ORDERS_IX_19_DSI O_IX_3
TPCC.ORDERS_IX_20_DSI O_IX_3
TPCC.ORDERS_IX_21_DSI O_IX_3
TPCC.ORDERS_IX_22_DSI O_IX_3
TPCC.ORDERS_IX_23_DSI O_IX_3
TPCC.ORDERS_IX_24_DSI O_IX_3
TPCC.ORDERS_IX_25_DSI O_IX_4
TPCC.ORDERS_IX_26_DSI O_IX_4
TPCC.ORDERS_IX_27_DSI O_IX_4
TPCC.ORDERS_IX_28_DSI O_IX_4
TPCC.ORDERS_IX_29_DSI O_IX_4
TPCC.ORDERS_IX_30_DSI O_IX_4
TPCC.ORDERS_IX_31_DSI O_IX_4
TPCC.ORDERS_IX_32_DSI O_IX_4
TPCC.ORDERS_IX_33_DSI O_IX_5
TPCC.ORDERS_IX_34_DSI O_IX_5
TPCC.ORDERS_IX_35_DSI O_IX_5
TPCC.ORDERS_IX_36_DSI O_IX_5

```
TPCC.ORDERS_IX_37_DSI O_IX_5
TPCC.ORDERS_IX_38_DSI O_IX_5
TPCC.ORDERS_IX_39_DSI O_IX_5
TPCC.ORDERS_IX_40_DSI O_IX_5
TPCC.ORDERS_IX_41_DSI O_IX_6
TPCC.ORDERS_IX_42_DSI O_IX_6
TPCC.ORDERS_IX_43_DSI O_IX_6
TPCC.ORDERS_IX_44_DSI O_IX_6
TPCC.ORDERS_IX_45_DSI O_IX_6
TPCC.ORDERS_IX_46_DSI O_IX_6
TPCC.ORDERS_IX_47_DSI O_IX_6
TPCC.ORDERS_IX_48_DSI O_IX_6
TPCC.ORDERS_IX_49_DSI O_IX_1
TPCC.ORDERS_IX_50_DSI O_IX_1
TPCC.ORDERS_IX_51_DSI O_IX_1
TPCC.ORDERS_IX_52_DSI O_IX_1
TPCC.ORDERS_IX_53_DSI O_IX_1
TPCC.ORDERS_IX_54_DSI O_IX_1
TPCC.ORDERS_IX_55_DSI O_IX_1
TPCC.ORDERS_IX_56_DSI O_IX_1
TPCC.ORDERS_IX_57_DSI O_IX_2
TPCC.ORDERS_IX_58_DSI O_IX_2
TPCC.ORDERS_IX_59_DSI O_IX_2
TPCC.ORDERS_IX_60_DSI O_IX_2
TPCC.ORDERS_IX_61_DSI O_IX_2
TPCC.ORDERS_IX_62_DSI O_IX_2
TPCC.ORDERS_IX_63_DSI O_IX_2
TPCC.ORDERS_IX_64_DSI O_IX_2
TPCC.ORDERS_IX_65_DSI O_IX_3
TPCC.ORDERS_IX_66_DSI O_IX_3
TPCC.ORDERS_IX_67_DSI O_IX_3
TPCC.ORDERS_IX_68_DSI O_IX_3
TPCC.ORDERS_IX_69_DSI O_IX_3
TPCC.ORDERS_IX_70_DSI O_IX_3
TPCC.ORDERS_IX_71_DSI O_IX_3
TPCC.ORDERS_IX_72_DSI O_IX_3
TPCC.ORDERS_IX_73_DSI O_IX_4
TPCC.ORDERS_IX_74_DSI O_IX_4
TPCC.ORDERS_IX_75_DSI O_IX_4
TPCC.ORDERS_IX_76_DSI O_IX_4
TPCC.ORDERS_IX_77_DSI O_IX_4
TPCC.ORDERS_IX_78_DSI O_IX_4
TPCC.ORDERS_IX_79_DSI O_IX_4
TPCC.ORDERS_IX_80_DSI O_IX_4
TPCC.ORDERS_IX_81_DSI O_IX_5
TPCC.ORDERS_IX_82_DSI O_IX_5
TPCC.ORDERS_IX_83_DSI O_IX_5
TPCC.ORDERS_IX_84_DSI O_IX_5
TPCC.ORDERS_IX_85_DSI O_IX_5
TPCC.ORDERS_IX_86_DSI O_IX_5
TPCC.ORDERS_IX_87_DSI O_IX_5
TPCC.ORDERS_IX_88_DSI O_IX_5
TPCC.ORDERS_IX_89_DSI O_IX_6
TPCC.ORDERS_IX_90_DSI O_IX_6
TPCC.ORDERS_IX_91_DSI O_IX_6
TPCC.ORDERS_IX_92_DSI O_IX_6
```

TPCC.ORDERS_IX_93_DSI O_IX_6
TPCC.ORDERS_IX_94_DSI O_IX_6
TPCC.ORDERS_IX_95_DSI O_IX_6
TPCC.ORDERS_IX_96_DSI O_IX_6
TPCC.NEWORDER_1_DSI NO_1
TPCC.NEWORDER_2_DSI NO_1
TPCC.NEWORDER_3_DSI NO_1
TPCC.NEWORDER_4_DSI NO_1
TPCC.NEWORDER_5_DSI NO_1
TPCC.NEWORDER_6_DSI NO_1
TPCC.NEWORDER_7_DSI NO_1
TPCC.NEWORDER_8_DSI NO_1
TPCC.NEWORDER_9_DSI NO_2
TPCC.NEWORDER_10_DSI NO_2
TPCC.NEWORDER_11_DSI NO_2
TPCC.NEWORDER_12_DSI NO_2
TPCC.NEWORDER_13_DSI NO_2
TPCC.NEWORDER_14_DSI NO_2
TPCC.NEWORDER_15_DSI NO_2
TPCC.NEWORDER_16_DSI NO_2
TPCC.NEWORDER_17_DSI NO_3
TPCC.NEWORDER_18_DSI NO_3
TPCC.NEWORDER_19_DSI NO_3
TPCC.NEWORDER_20_DSI NO_3
TPCC.NEWORDER_21_DSI NO_3
TPCC.NEWORDER_22_DSI NO_3
TPCC.NEWORDER_23_DSI NO_3
TPCC.NEWORDER_24_DSI NO_3
TPCC.NEWORDER_25_DSI NO_4
TPCC.NEWORDER_26_DSI NO_4
TPCC.NEWORDER_27_DSI NO_4
TPCC.NEWORDER_28_DSI NO_4
TPCC.NEWORDER_29_DSI NO_4
TPCC.NEWORDER_30_DSI NO_4
TPCC.NEWORDER_31_DSI NO_4
TPCC.NEWORDER_32_DSI NO_4
TPCC.NEWORDER_33_DSI NO_5
TPCC.NEWORDER_34_DSI NO_5
TPCC.NEWORDER_35_DSI NO_5
TPCC.NEWORDER_36_DSI NO_5
TPCC.NEWORDER_37_DSI NO_5
TPCC.NEWORDER_38_DSI NO_5
TPCC.NEWORDER_39_DSI NO_5
TPCC.NEWORDER_40_DSI NO_5
TPCC.NEWORDER_41_DSI NO_6
TPCC.NEWORDER_42_DSI NO_6
TPCC.NEWORDER_43_DSI NO_6
TPCC.NEWORDER_44_DSI NO_6
TPCC.NEWORDER_45_DSI NO_6
TPCC.NEWORDER_46_DSI NO_6
TPCC.NEWORDER_47_DSI NO_6
TPCC.NEWORDER_48_DSI NO_6
TPCC.NEWORDER_49_DSI NO_1
TPCC.NEWORDER_50_DSI NO_1
TPCC.NEWORDER_51_DSI NO_1
TPCC.NEWORDER_52_DSI NO_1

```
TPCC.NEWORDER_53_DSI NO_1
TPCC.NEWORDER_54_DSI NO_1
TPCC.NEWORDER_55_DSI NO_1
TPCC.NEWORDER_56_DSI NO_1
TPCC.NEWORDER_57_DSI NO_2
TPCC.NEWORDER_58_DSI NO_2
TPCC.NEWORDER_59_DSI NO_2
TPCC.NEWORDER_60_DSI NO_2
TPCC.NEWORDER_61_DSI NO_2
TPCC.NEWORDER_62_DSI NO_2
TPCC.NEWORDER_63_DSI NO_2
TPCC.NEWORDER_64_DSI NO_2
TPCC.NEWORDER_65_DSI NO_3
TPCC.NEWORDER_66_DSI NO_3
TPCC.NEWORDER_67_DSI NO_3
TPCC.NEWORDER_68_DSI NO_3
TPCC.NEWORDER_69_DSI NO_3
TPCC.NEWORDER_70_DSI NO_3
TPCC.NEWORDER_71_DSI NO_3
TPCC.NEWORDER_72_DSI NO_3
TPCC.NEWORDER_73_DSI NO_4
TPCC.NEWORDER_74_DSI NO_4
TPCC.NEWORDER_75_DSI NO_4
TPCC.NEWORDER_76_DSI NO_4
TPCC.NEWORDER_77_DSI NO_4
TPCC.NEWORDER_78_DSI NO_4
TPCC.NEWORDER_79_DSI NO_4
TPCC.NEWORDER_80_DSI NO_4
TPCC.NEWORDER_81_DSI NO_5
TPCC.NEWORDER_82_DSI NO_5
TPCC.NEWORDER_83_DSI NO_5
TPCC.NEWORDER_84_DSI NO_5
TPCC.NEWORDER_85_DSI NO_5
TPCC.NEWORDER_86_DSI NO_5
TPCC.NEWORDER_87_DSI NO_5
TPCC.NEWORDER_88_DSI NO_5
TPCC.NEWORDER_89_DSI NO_6
TPCC.NEWORDER_90_DSI NO_6
TPCC.NEWORDER_91_DSI NO_6
TPCC.NEWORDER_92_DSI NO_6
TPCC.NEWORDER_93_DSI NO_6
TPCC.NEWORDER_94_DSI NO_6
TPCC.NEWORDER_95_DSI NO_6
TPCC.NEWORDER_96_DSI NO_6
TPCC.NEWORDER_X_1_DSI NO_IX_1
TPCC.NEWORDER_X_2_DSI NO_IX_1
TPCC.NEWORDER_X_3_DSI NO_IX_1
TPCC.NEWORDER_X_4_DSI NO_IX_1
TPCC.NEWORDER_X_5_DSI NO_IX_1
TPCC.NEWORDER_X_6_DSI NO_IX_1
TPCC.NEWORDER_X_7_DSI NO_IX_1
TPCC.NEWORDER_X_8_DSI NO_IX_1
TPCC.NEWORDER_X_9_DSI NO_IX_2
TPCC.NEWORDER_X_10_DSI NO_IX_2
TPCC.NEWORDER_X_11_DSI NO_IX_2
TPCC.NEWORDER_X_12_DSI NO_IX_2
```

```
TPCC.NEWORDER_X_13_DSI NO_IX_2
TPCC.NEWORDER_X_14_DSI NO_IX_2
TPCC.NEWORDER_X_15_DSI NO_IX_2
TPCC.NEWORDER_X_16_DSI NO_IX_2
TPCC.NEWORDER_X_17_DSI NO_IX_3
TPCC.NEWORDER_X_18_DSI NO_IX_3
TPCC.NEWORDER_X_19_DSI NO_IX_3
TPCC.NEWORDER_X_20_DSI NO_IX_3
TPCC.NEWORDER_X_21_DSI NO_IX_3
TPCC.NEWORDER_X_22_DSI NO_IX_3
TPCC.NEWORDER_X_23_DSI NO_IX_3
TPCC.NEWORDER_X_24_DSI NO_IX_3
TPCC.NEWORDER_X_25_DSI NO_IX_4
TPCC.NEWORDER_X_26_DSI NO_IX_4
TPCC.NEWORDER_X_27_DSI NO_IX_4
TPCC.NEWORDER_X_28_DSI NO_IX_4
TPCC.NEWORDER_X_29_DSI NO_IX_4
TPCC.NEWORDER_X_30_DSI NO_IX_4
TPCC.NEWORDER_X_31_DSI NO_IX_4
TPCC.NEWORDER_X_32_DSI NO_IX_4
TPCC.NEWORDER_X_33_DSI NO_IX_5
TPCC.NEWORDER_X_34_DSI NO_IX_5
TPCC.NEWORDER_X_35_DSI NO_IX_5
TPCC.NEWORDER_X_36_DSI NO_IX_5
TPCC.NEWORDER_X_37_DSI NO_IX_5
TPCC.NEWORDER_X_38_DSI NO_IX_5
TPCC.NEWORDER_X_39_DSI NO_IX_5
TPCC.NEWORDER_X_40_DSI NO_IX_5
TPCC.NEWORDER_X_41_DSI NO_IX_6
TPCC.NEWORDER_X_42_DSI NO_IX_6
TPCC.NEWORDER_X_43_DSI NO_IX_6
TPCC.NEWORDER_X_44_DSI NO_IX_6
TPCC.NEWORDER_X_45_DSI NO_IX_6
TPCC.NEWORDER_X_46_DSI NO_IX_6
TPCC.NEWORDER_X_47_DSI NO_IX_6
TPCC.NEWORDER_X_48_DSI NO_IX_6
TPCC.NEWORDER_X_49_DSI NO_IX_1
TPCC.NEWORDER_X_50_DSI NO_IX_1
TPCC.NEWORDER_X_51_DSI NO_IX_1
TPCC.NEWORDER_X_52_DSI NO_IX_1
TPCC.NEWORDER_X_53_DSI NO_IX_1
TPCC.NEWORDER_X_54_DSI NO_IX_1
TPCC.NEWORDER_X_55_DSI NO_IX_1
TPCC.NEWORDER_X_56_DSI NO_IX_1
TPCC.NEWORDER_X_57_DSI NO_IX_2
TPCC.NEWORDER_X_58_DSI NO_IX_2
TPCC.NEWORDER_X_59_DSI NO_IX_2
TPCC.NEWORDER_X_60_DSI NO_IX_2
TPCC.NEWORDER_X_61_DSI NO_IX_2
TPCC.NEWORDER_X_62_DSI NO_IX_2
TPCC.NEWORDER_X_63_DSI NO_IX_2
TPCC.NEWORDER_X_64_DSI NO_IX_2
TPCC.NEWORDER_X_65_DSI NO_IX_3
TPCC.NEWORDER_X_66_DSI NO_IX_3
TPCC.NEWORDER_X_67_DSI NO_IX_3
TPCC.NEWORDER_X_68_DSI NO_IX_3
```

TPCC.NEWORDER_X_69_DSI NO_IX_3
TPCC.NEWORDER_X_70_DSI NO_IX_3
TPCC.NEWORDER_X_71_DSI NO_IX_3
TPCC.NEWORDER_X_72_DSI NO_IX_3
TPCC.NEWORDER_X_73_DSI NO_IX_4
TPCC.NEWORDER_X_74_DSI NO_IX_4
TPCC.NEWORDER_X_75_DSI NO_IX_4
TPCC.NEWORDER_X_76_DSI NO_IX_4
TPCC.NEWORDER_X_77_DSI NO_IX_4
TPCC.NEWORDER_X_78_DSI NO_IX_4
TPCC.NEWORDER_X_79_DSI NO_IX_4
TPCC.NEWORDER_X_80_DSI NO_IX_4
TPCC.NEWORDER_X_81_DSI NO_IX_5
TPCC.NEWORDER_X_82_DSI NO_IX_5
TPCC.NEWORDER_X_83_DSI NO_IX_5
TPCC.NEWORDER_X_84_DSI NO_IX_5
TPCC.NEWORDER_X_85_DSI NO_IX_5
TPCC.NEWORDER_X_86_DSI NO_IX_5
TPCC.NEWORDER_X_87_DSI NO_IX_5
TPCC.NEWORDER_X_88_DSI NO_IX_5
TPCC.NEWORDER_X_89_DSI NO_IX_6
TPCC.NEWORDER_X_90_DSI NO_IX_6
TPCC.NEWORDER_X_91_DSI NO_IX_6
TPCC.NEWORDER_X_92_DSI NO_IX_6
TPCC.NEWORDER_X_93_DSI NO_IX_6
TPCC.NEWORDER_X_94_DSI NO_IX_6
TPCC.NEWORDER_X_95_DSI NO_IX_6
TPCC.NEWORDER_X_96_DSI NO_IX_6
TPCC.STOCK_1_DSI S_1
TPCC.STOCK_2_DSI S_1
TPCC.STOCK_3_DSI S_1
TPCC.STOCK_4_DSI S_1
TPCC.STOCK_5_DSI S_2
TPCC.STOCK_6_DSI S_2
TPCC.STOCK_7_DSI S_2
TPCC.STOCK_8_DSI S_2
TPCC.STOCK_9_DSI S_3
TPCC.STOCK_10_DSI S_3
TPCC.STOCK_11_DSI S_3
TPCC.STOCK_12_DSI S_3
TPCC.STOCK_13_DSI S_4
TPCC.STOCK_14_DSI S_4
TPCC.STOCK_15_DSI S_4
TPCC.STOCK_16_DSI S_4
TPCC.STOCK_17_DSI S_5
TPCC.STOCK_18_DSI S_5
TPCC.STOCK_19_DSI S_5
TPCC.STOCK_20_DSI S_5
TPCC.STOCK_21_DSI S_6
TPCC.STOCK_22_DSI S_6
TPCC.STOCK_23_DSI S_6
TPCC.STOCK_24_DSI S_6
TPCC.STOCK_25_DSI S_1
TPCC.STOCK_26_DSI S_1
TPCC.STOCK_27_DSI S_1
TPCC.STOCK_28_DSI S_1

```
TPCC.STOCK_29_DSI S_2
TPCC.STOCK_30_DSI S_2
TPCC.STOCK_31_DSI S_2
TPCC.STOCK_32_DSI S_2
TPCC.STOCK_33_DSI S_3
TPCC.STOCK_34_DSI S_3
TPCC.STOCK_35_DSI S_3
TPCC.STOCK_36_DSI S_3
TPCC.STOCK_37_DSI S_4
TPCC.STOCK_38_DSI S_4
TPCC.STOCK_39_DSI S_4
TPCC.STOCK_40_DSI S_4
TPCC.STOCK_41_DSI S_5
TPCC.STOCK_42_DSI S_5
TPCC.STOCK_43_DSI S_5
TPCC.STOCK_44_DSI S_5
TPCC.STOCK_45_DSI S_6
TPCC.STOCK_46_DSI S_6
TPCC.STOCK_47_DSI S_6
TPCC.STOCK_48_DSI S_6
TPCC.ITEM_1_DSI I_1
TPCC.ORDERLIN_1_DSI OL_1
TPCC.ORDERLIN_2_DSI OL_1
TPCC.ORDERLIN_3_DSI OL_1
TPCC.ORDERLIN_4_DSI OL_1
TPCC.ORDERLIN_5_DSI OL_1
TPCC.ORDERLIN_6_DSI OL_1
TPCC.ORDERLIN_7_DSI OL_1
TPCC.ORDERLIN_8_DSI OL_1
TPCC.ORDERLIN_9_DSI OL_1
TPCC.ORDERLIN_10_DSI OL_1
TPCC.ORDERLIN_11_DSI OL_1
TPCC.ORDERLIN_12_DSI OL_1
TPCC.ORDERLIN_13_DSI OL_1
TPCC.ORDERLIN_14_DSI OL_1
TPCC.ORDERLIN_15_DSI OL_1
TPCC.ORDERLIN_16_DSI OL_1
TPCC.ORDERLIN_17_DSI OL_2
TPCC.ORDERLIN_18_DSI OL_2
TPCC.ORDERLIN_19_DSI OL_2
TPCC.ORDERLIN_20_DSI OL_2
TPCC.ORDERLIN_21_DSI OL_2
TPCC.ORDERLIN_22_DSI OL_2
TPCC.ORDERLIN_23_DSI OL_2
TPCC.ORDERLIN_24_DSI OL_2
TPCC.ORDERLIN_25_DSI OL_2
TPCC.ORDERLIN_26_DSI OL_2
TPCC.ORDERLIN_27_DSI OL_2
TPCC.ORDERLIN_28_DSI OL_2
TPCC.ORDERLIN_29_DSI OL_2
TPCC.ORDERLIN_30_DSI OL_2
TPCC.ORDERLIN_31_DSI OL_2
TPCC.ORDERLIN_32_DSI OL_2
TPCC.ORDERLIN_33_DSI OL_3
TPCC.ORDERLIN_34_DSI OL_3
TPCC.ORDERLIN_35_DSI OL_3
```

```
TPCC.ORDERLIN_36_DSI OL_3
TPCC.ORDERLIN_37_DSI OL_3
TPCC.ORDERLIN_38_DSI OL_3
TPCC.ORDERLIN_39_DSI OL_3
TPCC.ORDERLIN_40_DSI OL_3
TPCC.ORDERLIN_41_DSI OL_3
TPCC.ORDERLIN_42_DSI OL_3
TPCC.ORDERLIN_43_DSI OL_3
TPCC.ORDERLIN_44_DSI OL_3
TPCC.ORDERLIN_45_DSI OL_3
TPCC.ORDERLIN_46_DSI OL_3
TPCC.ORDERLIN_47_DSI OL_3
TPCC.ORDERLIN_48_DSI OL_3
TPCC.ORDERLIN_49_DSI OL_4
TPCC.ORDERLIN_50_DSI OL_4
TPCC.ORDERLIN_51_DSI OL_4
TPCC.ORDERLIN_52_DSI OL_4
TPCC.ORDERLIN_53_DSI OL_4
TPCC.ORDERLIN_54_DSI OL_4
TPCC.ORDERLIN_55_DSI OL_4
TPCC.ORDERLIN_56_DSI OL_4
TPCC.ORDERLIN_57_DSI OL_4
TPCC.ORDERLIN_58_DSI OL_4
TPCC.ORDERLIN_59_DSI OL_4
TPCC.ORDERLIN_60_DSI OL_4
TPCC.ORDERLIN_61_DSI OL_4
TPCC.ORDERLIN_62_DSI OL_4
TPCC.ORDERLIN_63_DSI OL_4
TPCC.ORDERLIN_64_DSI OL_4
TPCC.ORDERLIN_65_DSI OL_5
TPCC.ORDERLIN_66_DSI OL_5
TPCC.ORDERLIN_67_DSI OL_5
TPCC.ORDERLIN_68_DSI OL_5
TPCC.ORDERLIN_69_DSI OL_5
TPCC.ORDERLIN_70_DSI OL_5
TPCC.ORDERLIN_71_DSI OL_5
TPCC.ORDERLIN_72_DSI OL_5
TPCC.ORDERLIN_73_DSI OL_5
TPCC.ORDERLIN_74_DSI OL_5
TPCC.ORDERLIN_75_DSI OL_5
TPCC.ORDERLIN_76_DSI OL_5
TPCC.ORDERLIN_77_DSI OL_5
TPCC.ORDERLIN_78_DSI OL_5
TPCC.ORDERLIN_79_DSI OL_5
TPCC.ORDERLIN_80_DSI OL_5
TPCC.ORDERLIN_81_DSI OL_6
TPCC.ORDERLIN_82_DSI OL_6
TPCC.ORDERLIN_83_DSI OL_6
TPCC.ORDERLIN_84_DSI OL_6
TPCC.ORDERLIN_85_DSI OL_6
TPCC.ORDERLIN_86_DSI OL_6
TPCC.ORDERLIN_87_DSI OL_6
TPCC.ORDERLIN_88_DSI OL_6
TPCC.ORDERLIN_89_DSI OL_6
TPCC.ORDERLIN_90_DSI OL_6
TPCC.ORDERLIN_91_DSI OL_6
```

TPCC.ORDERLIN_92_DSI OL_6
TPCC.ORDERLIN_93_DSI OL_6
TPCC.ORDERLIN_94_DSI OL_6
TPCC.ORDERLIN_95_DSI OL_6
TPCC.ORDERLIN_96_DSI OL_6
TPCC.ORDERLIN_97_DSI OL_1
TPCC.ORDERLIN_98_DSI OL_1
TPCC.ORDERLIN_99_DSI OL_1
TPCC.ORDERLIN_100_DSI OL_1
TPCC.ORDERLIN_101_DSI OL_1
TPCC.ORDERLIN_102_DSI OL_1
TPCC.ORDERLIN_103_DSI OL_1
TPCC.ORDERLIN_104_DSI OL_1
TPCC.ORDERLIN_105_DSI OL_1
TPCC.ORDERLIN_106_DSI OL_1
TPCC.ORDERLIN_107_DSI OL_1
TPCC.ORDERLIN_108_DSI OL_1
TPCC.ORDERLIN_109_DSI OL_1
TPCC.ORDERLIN_110_DSI OL_1
TPCC.ORDERLIN_111_DSI OL_1
TPCC.ORDERLIN_112_DSI OL_1
TPCC.ORDERLIN_113_DSI OL_2
TPCC.ORDERLIN_114_DSI OL_2
TPCC.ORDERLIN_115_DSI OL_2
TPCC.ORDERLIN_116_DSI OL_2
TPCC.ORDERLIN_117_DSI OL_2
TPCC.ORDERLIN_118_DSI OL_2
TPCC.ORDERLIN_119_DSI OL_2
TPCC.ORDERLIN_120_DSI OL_2
TPCC.ORDERLIN_121_DSI OL_2
TPCC.ORDERLIN_122_DSI OL_2
TPCC.ORDERLIN_123_DSI OL_2
TPCC.ORDERLIN_124_DSI OL_2
TPCC.ORDERLIN_125_DSI OL_2
TPCC.ORDERLIN_126_DSI OL_2
TPCC.ORDERLIN_127_DSI OL_2
TPCC.ORDERLIN_128_DSI OL_2
TPCC.ORDERLIN_129_DSI OL_3
TPCC.ORDERLIN_130_DSI OL_3
TPCC.ORDERLIN_131_DSI OL_3
TPCC.ORDERLIN_132_DSI OL_3
TPCC.ORDERLIN_133_DSI OL_3
TPCC.ORDERLIN_134_DSI OL_3
TPCC.ORDERLIN_135_DSI OL_3
TPCC.ORDERLIN_136_DSI OL_3
TPCC.ORDERLIN_137_DSI OL_3
TPCC.ORDERLIN_138_DSI OL_3
TPCC.ORDERLIN_139_DSI OL_3
TPCC.ORDERLIN_140_DSI OL_3
TPCC.ORDERLIN_141_DSI OL_3
TPCC.ORDERLIN_142_DSI OL_3
TPCC.ORDERLIN_143_DSI OL_3
TPCC.ORDERLIN_144_DSI OL_3
TPCC.ORDERLIN_145_DSI OL_4
TPCC.ORDERLIN_146_DSI OL_4
TPCC.ORDERLIN_147_DSI OL_4

TPCC.ORDERLIN_148_DSI OL_4
TPCC.ORDERLIN_149_DSI OL_4
TPCC.ORDERLIN_150_DSI OL_4
TPCC.ORDERLIN_151_DSI OL_4
TPCC.ORDERLIN_152_DSI OL_4
TPCC.ORDERLIN_153_DSI OL_4
TPCC.ORDERLIN_154_DSI OL_4
TPCC.ORDERLIN_155_DSI OL_4
TPCC.ORDERLIN_156_DSI OL_4
TPCC.ORDERLIN_157_DSI OL_4
TPCC.ORDERLIN_158_DSI OL_4
TPCC.ORDERLIN_159_DSI OL_4
TPCC.ORDERLIN_160_DSI OL_4
TPCC.ORDERLIN_161_DSI OL_5
TPCC.ORDERLIN_162_DSI OL_5
TPCC.ORDERLIN_163_DSI OL_5
TPCC.ORDERLIN_164_DSI OL_5
TPCC.ORDERLIN_165_DSI OL_5
TPCC.ORDERLIN_166_DSI OL_5
TPCC.ORDERLIN_167_DSI OL_5
TPCC.ORDERLIN_168_DSI OL_5
TPCC.ORDERLIN_169_DSI OL_5
TPCC.ORDERLIN_170_DSI OL_5
TPCC.ORDERLIN_171_DSI OL_5
TPCC.ORDERLIN_172_DSI OL_5
TPCC.ORDERLIN_173_DSI OL_5
TPCC.ORDERLIN_174_DSI OL_5
TPCC.ORDERLIN_175_DSI OL_5
TPCC.ORDERLIN_176_DSI OL_5
TPCC.ORDERLIN_177_DSI OL_6
TPCC.ORDERLIN_178_DSI OL_6
TPCC.ORDERLIN_179_DSI OL_6
TPCC.ORDERLIN_180_DSI OL_6
TPCC.ORDERLIN_181_DSI OL_6
TPCC.ORDERLIN_182_DSI OL_6
TPCC.ORDERLIN_183_DSI OL_6
TPCC.ORDERLIN_184_DSI OL_6
TPCC.ORDERLIN_185_DSI OL_6
TPCC.ORDERLIN_186_DSI OL_6
TPCC.ORDERLIN_187_DSI OL_6
TPCC.ORDERLIN_188_DSI OL_6
TPCC.ORDERLIN_189_DSI OL_6
TPCC.ORDERLIN_190_DSI OL_6
TPCC.ORDERLIN_191_DSI OL_6
TPCC.ORDERLIN_192_DSI OL_6
TPCC.HISTORY_1_DSI H_1
TPCC.HISTORY_2_DSI H_1
TPCC.HISTORY_3_DSI H_1
TPCC.HISTORY_4_DSI H_1
TPCC.HISTORY_5_DSI H_1
TPCC.HISTORY_6_DSI H_1
TPCC.HISTORY_7_DSI H_1
TPCC.HISTORY_8_DSI H_1
TPCC.HISTORY_9_DSI H_2
TPCC.HISTORY_10_DSI H_2
TPCC.HISTORY_11_DSI H_2

TPCC.HISTORY_12_DSI H_2
TPCC.HISTORY_13_DSI H_2
TPCC.HISTORY_14_DSI H_2
TPCC.HISTORY_15_DSI H_2
TPCC.HISTORY_16_DSI H_2
TPCC.HISTORY_17_DSI H_3
TPCC.HISTORY_18_DSI H_3
TPCC.HISTORY_19_DSI H_3
TPCC.HISTORY_20_DSI H_3
TPCC.HISTORY_21_DSI H_3
TPCC.HISTORY_22_DSI H_3
TPCC.HISTORY_23_DSI H_3
TPCC.HISTORY_24_DSI H_3
TPCC.HISTORY_25_DSI H_4
TPCC.HISTORY_26_DSI H_4
TPCC.HISTORY_27_DSI H_4
TPCC.HISTORY_28_DSI H_4
TPCC.HISTORY_29_DSI H_4
TPCC.HISTORY_30_DSI H_4
TPCC.HISTORY_31_DSI H_4
TPCC.HISTORY_32_DSI H_4
TPCC.HISTORY_33_DSI H_5
TPCC.HISTORY_34_DSI H_5
TPCC.HISTORY_35_DSI H_5
TPCC.HISTORY_36_DSI H_5
TPCC.HISTORY_37_DSI H_5
TPCC.HISTORY_38_DSI H_5
TPCC.HISTORY_39_DSI H_5
TPCC.HISTORY_40_DSI H_5
TPCC.HISTORY_41_DSI H_6
TPCC.HISTORY_42_DSI H_6
TPCC.HISTORY_43_DSI H_6
TPCC.HISTORY_44_DSI H_6
TPCC.HISTORY_45_DSI H_6
TPCC.HISTORY_46_DSI H_6
TPCC.HISTORY_47_DSI H_6
TPCC.HISTORY_48_DSI H_6
TPCC.HISTORY_49_DSI H_1
TPCC.HISTORY_50_DSI H_1
TPCC.HISTORY_51_DSI H_1
TPCC.HISTORY_52_DSI H_1
TPCC.HISTORY_53_DSI H_1
TPCC.HISTORY_54_DSI H_1
TPCC.HISTORY_55_DSI H_1
TPCC.HISTORY_56_DSI H_1
TPCC.HISTORY_57_DSI H_2
TPCC.HISTORY_58_DSI H_2
TPCC.HISTORY_59_DSI H_2
TPCC.HISTORY_60_DSI H_2
TPCC.HISTORY_61_DSI H_2
TPCC.HISTORY_62_DSI H_2
TPCC.HISTORY_63_DSI H_2
TPCC.HISTORY_64_DSI H_2
TPCC.HISTORY_65_DSI H_3
TPCC.HISTORY_66_DSI H_3
TPCC.HISTORY_67_DSI H_3

TPCC.HISTORY_68_DSI H_3
TPCC.HISTORY_69_DSI H_3
TPCC.HISTORY_70_DSI H_3
TPCC.HISTORY_71_DSI H_3
TPCC.HISTORY_72_DSI H_3
TPCC.HISTORY_73_DSI H_4
TPCC.HISTORY_74_DSI H_4
TPCC.HISTORY_75_DSI H_4
TPCC.HISTORY_76_DSI H_4
TPCC.HISTORY_77_DSI H_4
TPCC.HISTORY_78_DSI H_4
TPCC.HISTORY_79_DSI H_4
TPCC.HISTORY_80_DSI H_4
TPCC.HISTORY_81_DSI H_5
TPCC.HISTORY_82_DSI H_5
TPCC.HISTORY_83_DSI H_5
TPCC.HISTORY_84_DSI H_5
TPCC.HISTORY_85_DSI H_5
TPCC.HISTORY_86_DSI H_5
TPCC.HISTORY_87_DSI H_5
TPCC.HISTORY_88_DSI H_5
TPCC.HISTORY_89_DSI H_6
TPCC.HISTORY_90_DSI H_6
TPCC.HISTORY_91_DSI H_6
TPCC.HISTORY_92_DSI H_6
TPCC.HISTORY_93_DSI H_6
TPCC.HISTORY_94_DSI H_6
TPCC.HISTORY_95_DSI H_6
TPCC.HISTORY_96_DSI H_6
::::::::::::::
cw_env.comp
::::::::::::::
TPCC,TPCC_SCHEMA,ITEM,STOCK; <<< TPCC ,ITEM, STOCK >>>
1,4,0,I_IM_ID,I_NAME,I_PRICE,I_DATA,1,1,100000,100000; <<< STAY_ALL, CACHE_OFF, NO_COMP >>>
0,1,0,S_QUANTITY,2,1,2112,2112,100000,100000; <<< 1 COLUMN, CACHE_OFF, NO_COMP >>>
::::::::::::::
fssqlenv
::::::::::::::
;
; All Rights Reserved, Copyright(c) FUJITSU 1993, 1994, 1995
; All Rights Reserved, Copyright(c) PFU   1993, 1994, 1995
;
; :
;
;
;   : ';'
;   :
;   : 11024
;
;;;
;MAX_CONNECT_TCP=(260)
MAX_CONNECT_TCP=(100)
;MAX_CONNECT_SYS=(250)
;MAX_CONNECT_SYS=(470)
MAX_CONNECT_SYS=(50)
; rdbsysconfig -> RDBCNTNUM

```
JOIN_RULE=(F,3)
COMMUNICATION_BUFFER=(8)
;INCLUSION_TYPE=NEW
;MAX_PARALLEL=20
RDB2_TCP_LEVEL=(LEVEL2)
;SOC_SELECT_TIME=(20000)
;SOC_SELECT_TIME=(5000)
SOC_SELECT_TIME=(10000)
SOC_WRITE_SIZE=(8)
;DEBUG_INFO = ON,2084


:::::::::::::
rdbbuf
:::::::::::::
#
# All Rights Reserved, Copyright(c) FUJITSU 1993, 1994, 1995
# All Rights Reserved, Copyright(c) PFU     1993, 1994, 1995
#
# :
#
#
#    : '#'
#       :
#    : 11024
#
#   << >>
#            =
#
#######################################################################nrk
BUFFER1K  = 64
BUFFER2K  = 64
BUFFER4K  = 512
BUFFER8K  = 384
BUFFER16K = 32
BUFFER32K = 32


:::::::::::::
rdbpool
:::::::::::::
#
# All Rights Reserved, Copyright(c) FUJITSU 1993, 1994, 1995, 1996
# All Rights Reserved, Copyright(c) PFU     1993, 1994, 1995, 1996
#
# :
#
#
#    : '#'
#       :
#    : 11024
#
#   << >>
#            = , ,
#
#######################################################################nrk
# system
```

```
#      =      #
#-------------------------------------------------------------------------
ARC_ALCT   = 0       ,1    ,1024000000 #508
BCM_BPC    = 124928  ,4096 ,1024000000 #508
BCM_EEXT   = 310272  ,8192 ,1024000000 #1532
BCM_ESUB   = 351232  ,1024 ,1024000000 #60
BCM_GPCT   = 4096    ,4096 ,1024000000 #508
BCM_IOPROC = 78848   ,4096 ,1024000000 #380
BCM_LOGAREA = 0      ,1    ,1024000000 #1020
BCM_LOGLIST = 0      ,1    ,1024000000 #252
BCM_PGC    = 249856  ,4096 ,1024000000 #1020
BCM_WKACC  = 0       ,1024 ,1024000000 #252
BCM_WKDMON = 256     ,1024 ,1024000000 #252
BCM_WKSPC  = 0       ,1024 ,1024000000 #60
BCM_WKSSPC = 0       ,1024 ,1024000000 #124
CCR_COMINF = 2629632 ,16384 ,1024000000 #3836(0xefc)
CCR_FGRP   = 63488   ,4096 ,1024000000 #252
CCR_IDT    = 848896  ,4096 ,1024000000 #912(0x390)
CCR_KAIOCB = 271360  ,1024 ,1024000000 #92
CCR_LWPIDT = 15360   ,4096 ,1024000000 #112(0x70)
CCR_POLMCTL = 2048000 ,16384 ,1024000000 #4336(0x10+0x50*(42+12))
CCR_SANQUE = 29696   ,1024 ,1024000000 #64
CCR_USRCON = 412672  ,4096 ,1024000000 #448(0x1c0)
CCR_USRSTK = 34030202 ,65544 ,1024000000 #65536
CCR_WLIST  = 22528   ,1024 ,1024000000 #28(0x1c)
CCR_WPID   = 306288  ,1024 ,1024000000 #60
DSM_DSAH   = 786432  ,4096 ,1024000000 #2044
DSM_DSAP   = 13578240 ,1024000 ,1024000000 #124
DSM_DSIL   = 3085312 ,1024 ,1024000000 #60
DSM_DSVP   = 7168    ,1024 ,1024000000 #60
DSM_DSVQ   = 61440   ,1024 ,1024000000 #92
DSM_DSWH   = 0       ,4096 ,1024000000 #2044
DSM_DSWP   = 0       ,1024 ,1024000000 #124
DSM_DUSI   = 16384   ,1024 ,1024000000 #60
DSM_DWFL   = 0       ,1024 ,1024000000 #60
DSM_DWUI   = 0       ,1024 ,1024000000 #60
LCM_LOGCNTL = 63488  ,4096 ,1024000000 #252
SCI_CMD    = 73728   ,4096 ,1024000000 #508
SCI_CONBF  = 4096    ,4096 ,1024000000 #508
SSV_IINF   = 31774   ,1024 ,1024000000 #124
TCM_TRAN   = 63488   ,4096 ,1024000000 #252
UTY_UNQUE  = 256     ,1024 ,1024000000 #172
UTY_UNDB   = 0       ,4096 ,1024000000 #508
UTY_UNDSI  = 0       ,1024 ,1024000000 #124
XCM_KHASH  = 0       ,4096 ,1024000000 #1036
XCM_KMEM   = 0       ,4096 ,1024000000 #2044
XCM_KQUE   = 0       ,1024 ,1024000000 #28
XCM_KTERM  = 0       ,1024 ,1024000000 #28
XCM_LOCK   = 1024000 ,10240 ,1024000000 #60
XCM_LPHASH = 3539558 ,4096 ,1024000000 #1028
XCM_NLOWN  = 5120    ,1024 ,1024000000 #28
XCM_NLQUE  = 10240   ,10240 ,1024000000 #60
XCM_NLRSC  = 10240   ,1024 ,1024000000 #252
XCM_OWNER  = 22528   ,1024 ,1024000000 #124
XCM_QUE    = 3379200 ,102400 ,1024000000 #124
XCM_TTERM  = 841728  ,1024 ,1024000000 #44
```

```
XCM_WQUE_S  = 0      ,1024 ,1024000000  #76
XCM_RSC_S   = 0      ,1024 ,1024000000  #60
#-------------------------------------------------------------------------
# group
#     =       #
#-------------------------------------------------------------------------
BCM_DFPOOL_G = 128   ,1024 ,1024000000  #124
BCM_DPCT_G  = 64     ,1024 ,1024000000  #60
#CCR_GCOMINF = 32768 ,16384 ,1024000000  #3836(0xefc)
CCR_GCOMINF = 8192   ,16384 ,1024000000  #3836(0xefc)
XCM_BITMAP_G = 96    ,1024 ,1024000000  #92
XCM_BITMNG_G = 64    ,1024 ,1024000000  #60
#XCM_RSC_G   = 64    ,1024 ,1024000000  #60
XCM_RSC_G   = 3520   ,1024 ,1024000000  #60
#XCM_WQUE_G  = 80    ,1024 ,1024000000  #76
XCM_WQUE_G  = 20000  ,1024 ,1024000000  #76

#-------------------------------------------------------------------------
# local
#     =       #
#-------------------------------------------------------------------------
BCM_DFPOOL  = 128    ,1024 ,1024000000  #124
BCM_DPCT    = 64     ,1024 ,1024000000  #60
#BCM_LPCT   = 64     ,1024 ,1024000000  #60
BCM_LPCT    = 960    ,1024 ,1024000000  #60
#BCM_LPG    = 256    ,1024 ,1024000000  #252
BCM_LPG     = 256    ,270000 ,1024000000  #252
BCM_PFT     = 256    ,67000 ,1024000000  #252
CCR_LCOMINF = 8240   ,307200 ,1024000000  #3836(0xefc)
#CCR_LCOMINF = 8192  ,40960 ,1024000000  #3836(0xefc)
DSM_DDSF    = 256    ,1024 ,1024000000  #252
##DSM_DDSF  = 411136 ,1024 ,1024000000  #252
DSM_DESF    = 256    ,1024 ,1024000000  #252
##DSM_DESF  = 736000 ,1024 ,1024000000  #252
SAP_KEY     = 4096   ,16384 ,1024000000  #4092
SCI_CS      = 0      ,1    ,1024000000  #124
#XCM_BITMAP = 96     ,1024 ,1024000000  #92
XCM_BITMAP  = 96     ,20000 ,1024000000  #92
XCM_BITMNG  = 64     ,2400 ,1024000000  #60
XCM_RSC     = 64     ,1024 ,1024000000  #60
XCM_WQUE    = 80     ,1024 ,1024000000  #76
XCM_THASH   = 48     ,1400 ,1024000000  #44
#XCM_TQUE   = 80     ,1024 ,1024000000  #76
XCM_TQUE    = 400    ,1024 ,1024000000  #76


::::::::::::::
rdbsysconfig
::::::::::::::
#
# All Rights Reserved, Copyright(c) FUJITSU 1993, 1994, 1995, 1999
# All Rights Reserved, Copyright(c) PFU    1993, 1994, 1995
#
#####################################################################nrk
RDBDIRSPACE1=/export/home/RDBDIR4
RDBDIRSPACE2=/export/home/RDBDIR4
#RDBLOG=512, 512
RDBLOG=1024, 1024
```

```
RDBCORE=/rdbptc/RDBCORE #directory name
RDBCNTNUM=50        #for cl-sv & 3tier #fssqlenv -> MAX_CONECT_SYS
#RDBCNTNUM=220      #fssqlenv -> MAX_CONECT_SYS
RDBPRJCODE=0xdb

RDBSYSBUF=/rdbptc/tpcc80/SYS
RDBSQLENV=/rdbptc/tpcc80/SYS/fssqlenv
RDBLOGMANAGE=/rdbptc/tpcc80/SYS
RDBPOOLCFG=/rdbptc/tpcc80/SYS


#----------------------------------------------------------------
#RDBFIXBUFMEM=80         #for LOAD
#RDBFIXBUFMEMADDR=0x36000000

#RDBFIXBUFMEM=2752     # 991010 OK
#--
RDBFIXBUFMEM=2816   #2896 #2928 #2944 #2912 #2848

#RDBFIXBUFMEMADDR=0x40000000 # 991010 OK
#RDBFIXBUFMEMADDR=0x37000000 #0x34000000 #0x35000000 #0x3a000000
RDBFIXBUFMEMADDR=0x3c000000

#-----------------------------------------------------
RDBEXTMEMADDR=0xed000000
RDBEXTMEM=8192

####################################################################
RDBPRESCHED=10
RDBKCHKSKIPCNT=200   #
#RDBKCHKSKIPCNT=300  #
RDBKTAJUUDOSDP = 2016
RDBKLISTNUMSDP = 5
RDBKCATENUMSDP=252
#RDBKAIOREP = 6
RDBKAIOREP = 12  #980531
#RDBKAIOREP = 18  #980531
##RDBKAIOSAV = 30
RDBKAIOYLD = 0

RDBKAIOCNT    = yes
RDBKAIOD9F    = yes
RDBKAIOSELFWAIT   = yes
RDBKAIODSP    = yes
#RDBIOERRDOWN = yes
####################################################################nrk
##RDBVER: UXP/DS_RDBII_V20L21_3/22_version
####################################################################
RDBSDPCPU = 0,1,2,3

RDBCCRDMCPU=0
RDBRECEPCPU=0
#---
RDBSORTCPU=0
RDBTCPIPCPU=0
RDBALFCPU=0  #990610 5 --> 1 --> 5
```

```
RDBDBSCPU=0  #NG  1,2,3      ##
RDBDIRCPU=0
#---
RDBIOCPU=0
RDBTLFCPU=1  #5.4i  #0
#---
RDBWKSCPU=0


####I/O
RDBREADUNC = NO        #

RDBDBSNUM+ = 350        #

RDBMAXLWP =  100      #98.11.11 oza
#RDBMAXDBIO = 20        #
RDBMAXDBIO = 8  #6  #NG3 #6good #12 #14ng  #12ok #5NG #10  # 990309 oza
RDBMAXRCPIO = 9 #5 #7 #?2  #7?  #13 #15?  #13ok #6NG #11 # old=20 RCP
RDBNEWPAGE = 1        # LRU
#RDBNEWPAGE = 2        # LRU
RDBANTIQUEPAGE = 4 # LRU
RDBIOUNITNUM = 1,1
RDBSORTUNITNUM = 1,1

##LOG
#RDBLOGAIONUM=32 #
RDBLOGAIONUM=128 #
RDBLOGBIONUM=256 #
RDBLOGIOSLEEP=20 #
RDBLOGSLTRNUM= 2  #
RDBLOGGRCOMMIT=4  #5  #6 #4 #

#RDBKTAJUUDOSDP=5
#RDBKCATENUMSDP=5
#RDBKCATENUMSDP=60

##SLK/LWP
RDBSLKLOOP=10  #
#RDBLWPLOOP=100 #
RDBSEMMODE=IPC  #

##my_mutex
#RDBDBGSLKCNT=yes,yes
#RDBDBGSLKCNT=yes
#RDBDBGMUTCNT=yes
#RDBDBGWPCCNT = yes
#RDBDBGWAITPOS = yes
#RDBKAIOCNT   = yes

RDBIXSECDWD=YES     #
RDBSDPLDBALMODE = 2

:::::::::::::::
rdbsysparm
:::::::::::::::
#
```

```
# All Rights Reserved, Copyright(c) FUJITSU 1996
# All Rights Reserved, Copyright(c) PFU     1996
#
# Title: RDB system definition file
#
############################################################################
# DO NOT TOUCH ME!!
#
#RDBMEMBLKSIZE=32
#RDBMEMBLKSIZE=64
RDBMEMBLKSIZE=128
RDBLBUFSIZE=0,128,512
COMMUNICATION_BUFFER=1
SORT_MEM_SIZE=64
WORK_MEM_SIZE=64
CGP_INIT_SIZE=1
###CGP_ELEM=10
MEM_CMD_POOL_SIZE=14 #1
#MEM_LC1_POOL_SIZE=1
#MEM_LC2_POOL_SIZE=1
MEM_LC3_POOL_SIZE=28  #1
MEM_OPL_POOL_SIZE=148  #1
#MEM_OPT_POOL_SIZE=1
MEM_SCT_POOL_SIZE=10  #1
#MEM_SPL_POOL_SIZE=1
DYN_SQL_BUFFER=3, 1, 3
TID_BUFFER=1, 1, 3
CURSOR_NAME_BUFFER=1, 1, 1
BUFFER_SIZE=1, 1
RESULT_BUFFER=0, 1
OPL_BUFFER_SIZE=1
MAX_CONNECT_SYS=20
DESC_NUM=256
:::::::::::::
sh.crbuf.mk
:::::::::::::
#!/bin/csh
rm sh.crbuf crbuf.aa
echo "# 1760WH xxxMB###########################" >sh.crbuf
@ D= 50
@ W= 50
@ T= 0
@ XK= 0

echo "#ORDERLINE(M)----------------------------------------">>sh.crbuf
#@ X=4500 #6000
@ X=6000  # 6 bunkatsu 1864WH
@ X=6800  # 6 bunkatsu 2112WH
@ X=6620  # 6 bunkatsu 2112WH
#@ A=$X - $D ; @ S=$A - $W; @ T= 0
@ A=$X - 60 ; @ S=$A - 60; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A $A  -S  $S -x -l 1 -m 5 OL_$P  32K $X">>sh.crbuf
 @ T= $T + $X * 32
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#OL $TM $XM"
```

```
echo "# STOCK---------------------------------"  >>sh.crbuf
#@ X=12750 + 6883 + 3300 + 2530
#@ X=22633 # 9->8bunkatsu 991111
#@ X=25866 # 8->7bunkatsu 991111
#@ X=32738 # 8->7bunkatsu 991111
#@ X=21000 #22619  #22916 # 10 bunkatsu 991120
#@ X=30000  # 7 bunkatsu 1864WH
# @ X=10000  # 7 bunkatsu 2090WH
# @ X=33500 = NG
@ X=33000  # 6 bunkatsu 2112WH
@ X=32150  # 6 bunkatsu 2112WH
@ X=25900  # 6 bunkatsu 2112WH

#@ A=$X - 400 ; @ S=$A - 100; @ T= 0
#@ A=$X - 500 ; @ S=$A - 150; @ T= 0   # for 7-bunkatsu
#@ A=$X - 1000 ; @ S=$A - 150; @ T= 0   # for 7-bunkatsu
#@ A=$X - 700 ; @ S=$A - 220; @ T= 0   # for 10-bun,1760WH
#@ A=$X - 700 ; @ S=$A - 250; @ T= 0   # for 7-bun,1848WH
#@ A=$X - 600 ; @ S=$A - 400; @ T= 0   # for 6-bun 2112WH 5.4c
@ A=$X - 800 ; @ S=$A - 400; @ T= 0   # 5.4g

#echo "$X $A $S"
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A $A -S $S -x -l 1 -m 3 S_$P  4K $X ">>sh.crbuf
 @ T= $T + $X * 4
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#S $TM $XM"

echo "#NEWORDER(M)-----------------------------------">>sh.crbuf
@ X=4000 #3000 #3124 #4166
@ A=$X - $D ; @ S=$A - $W; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A  $A -S  $S -x -l 1 -m 5 NO_$P  8K $X">>sh.crbuf
 @ T= $T + $X * 8
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#N $TM $XM"

echo "#NEWORDER_IX(MB)----------------------------------">>sh.crbuf
@ X=3661 #2746 #3024 #4166
@ X=4086
@ X=4800
@ A=$X - $D ; @ S=$A - $W; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A  $A -S  $S -x -l 1 -m 5 NO_IX_$P  8K $X">>sh.crbuf
 @ T= $T + $X * 8
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#NX $TM $XM"

echo "# CUSTOMER----------------------------------------">>sh.crbuf
@ X=2666 #2000 #1450 #1800
@ A=$X - 180 ; @ S=$A - 150; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A $A  -S $S -x -l 2 -m 4 C_$P  4K $X">>sh.crbuf
 @ T= $T + $X * 8
end
```

```
@ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#C $TM $XM"

echo "# CUSTOMER_IX(M)------------------------------------">>sh.crbuf
@ X=1100 #825 #1100
@ A=$X - $D ; @ S=$A - $W; @ T= 0
foreach P ( 1 2 3 4 5 6  )
 echo "rdbcrbf -A $A  -S $S -x -l 1 -m 5 C_IX_$P  16K $X">>sh.crbuf
 @ T= $T + $X * 16
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#CX $TM $XM"

echo "# ORDERS(M)---------------------------------------">>sh.crbuf
@ X=1432 #1074 #999 #1333
@ A=$X - 75 ; @ S=$A - 75; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A $A  -S  $S -x -l 1 -m 5 O_$P  8K $X">>sh.crbuf
 @ T= $T + $X * 8
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#O $TM $XM"

echo "# ORDER_IX(M)--------------------------------------">>sh.crbuf
@ X=3820 #2865 #2805 #1668 #2225
#@ A=$X - $D ; @ S=$A - $W; @ T= 0
@ A=$X - 80 ; @ S=$A - 80; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A $A  -S  $S -x -l 1 -m 5 O_IX_$P  8K $X">>sh.crbuf
 @ T= $T + $X * 8
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#OX $TM $XM"

echo "#ORDERLINE(M)-------------------------------------">>sh.crbuf
@ X=130 #98 #178  #238
@ A=$X - 25 ; @ S=$A - 5; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A $A  -S  $S -x -l 1 -m 5 OL_$P  8K $X">>sh.crbuf
 @ T= $T + $X * 8
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#OL2 $TM $XM"

echo "#HISTORY(M)-------------------------------------">>sh.crbuf
@ X=466 #350
@ A=$X - $D ; @ S=$A - $W; @ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "rdbcrbf -A  $A -S  $S -x -l 1 -m 5 H_$P  4K $X">>sh.crbuf
 @ T= $T + $X * 4
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#H $TM $XM"

#=====================================================================
echo "                        ">crbuf.aa

#echo "#D(M)-----------------------------------"

@ X=3520 #4400 #3520
foreach P ( 1 2 3 4 5 6 )
 echo "D_$P 1K $X 100 100 1 0 fixed">>crbuf.aa
```

```
 @ T= $T + $X * 1
end

 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#D $TM $XM"

#echo "#I(M)-----------------------------------"
@ X=14350 #14300
@ T= $X * 1
echo "I_1 1K $X 100 100 1 0 fixed" >>crbuf.aa
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#I $TM $XM"

#echo "#CX2(M)-----------------------------------"
@ X=14 #11 #10
@ X=16 # 2112WH
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "C_IX_$P 32K $X 100 100 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 32
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#CX2 $TM $XM"

#echo "#OX2(M)-----------------------------------"
@ X=42 #34 #30 #48
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "O_IX_$P 32K $X 100 100 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 32
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#OX2 $TM $XM"

#echo "#NOX2(M)-----------------------------------"
@ X=20 #11 #10 #16
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "NO_IX_$P 32K $X 100 100 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 32
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#NOX2 $TM $XM"


#echo "#W(M)-----------------------------------"
@ X=352 #440
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "W_$P 1K $X 100 100 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 1
end
 @ XK= $XK + $T ; @ XM= $XK / 1024; @ TM= $T / 1024; echo "#W $TM $XM"

#echo "#C2(M)-----------------------------------"
@ X=3
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "C_$P 1K $X 100 100 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 1
end
```

```
@ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#C2 $TM $XM"

#echo "#O2(M)-----------------------------------"
@ X=3
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "O_$P 1K $X 100 100 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 1
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#O2 $TM $XM"

#echo "#NO2(M)-----------------------------------"
@ X=10
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "NO_$P 1K $X 80 50 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 1
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#NO2 $TM $XM"

#echo "#S2(M)-----------------------------------"
@ X=3
@ T= 0
foreach P ( 1 2 3 4 5 6 )
 echo "S_$P 1K $X 80 50 1 0 fixed">>crbuf.aa
 @ T= $T + $X * 1
end
 @ XK= $XK + $T ; @ XM= $XK ⁄ 1024; @ TM= $T ⁄ 1024; echo "#S2 $TM $XM"

:::::::::::::::
sqlenv.tcp
:::::::::::::::
SERVER_SPEC = ( RDB2_TCP, SV1 ,TPCC , slice, 2001 )
DEFAULT_CONNECTION = ( SV1,TPCC,TPCC )
TRAN_SPEC = ( TRANSACTION_ROLLBACK )
DESCRIPTOR_SPEC = ( 30,1 )
;BUFFER_SIZE = ( 16 )
WAIT_TIME = ( 0 )
NCHAR_CODE = ( EUC )
OPL_BUFFER_SIZE = ( 280 )
;;;CHARACTER_TRANSLATE = CLIENT
CHAR_CODE = EUC
RESULT_BUFFER = ( 4,1 )
;SQL_SNAP = (ON,⁄risu02/tpcc.tcp.snap.970206,2 )
;MSG_PRINT = ( ON )
DSO_LOCK = (TPCC.ORDERLINE_DSO⁄EX,TPCC.HISTORY_DSO⁄EX,TPCC.CUSTOMER_IX_DSO⁄SH,
        TPCC.ITEM_DSO⁄SH,TPCC.NEWORDER_DSO⁄EX,TPCC.ORDERS_DSO⁄EX,
        TPCC.NEWORDER_IX_DSO⁄EX,TPCC.ORDERS_IX_DSO⁄EX)
;DSO_LOCK = ( TPCC.CUSTOMER_IX_DSO⁄SH,TPCC.ITEM_DSO⁄SH )
SIGNAL_INF = NO
SORT_MEM_SIZE = 128
WORK_MEM_SIZE = 64
```

Tuxedo Configuration values

```
#::::::::::::::
ubbconfig.c1
::::::::::::::
#
#      ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY       40001  # IPC KEY from 32,768 to 16,777,215
MASTER       c1     # machine on which master copy is found
UID          30     # user id as displayed by command "id"
GID          5433   # group id as displayed by command "id"
PERM         0666   # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS  3700# max no of processes accesing bulleting board
MAXGTT       1000   # maximum simultaneous global transactions
MAXSERVERS   200    # maximum number of servers
MAXSERVICES  3100   # 991022 to big OK?.old=200 #maximum number of services
MAXCONV      1
MODEL        SHM    # SHM=single processor, MP=multi processor
LDBAL        N      # load balancing, Y=yes, N=no
CMTRET       COMPLETE
#MAXBUFTYPE  16     # maximum buffer types
#MAXBUFSTYPE 32     # maximum buffer subtypes
SCANUNIT     30     # scan program wake-up time in secs.
SANITYSCAN   5      # sanity scan wake-up
DBBLWAIT     1      # scanunit multiplier for DBBL max time wait
BBLQUERY     60     # check out wake-up time
BLOCKTIME    10     # blocking call time-out
NOTIFY       DIPIN
SYSTEM_ACCESS  FASTPATH
USIGNAL      SIGUSR2


*MACHINES
"c1"   LMID="c1"
      TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c1"
      ROOTDIR="/export/home/tuxedo"
      APPDIR="/export/home/dbbench/tuxedo"
      ULOGPFX="/export/home/dbbench/tuxedo/ULOGc1"
      ENVFILE="/export/home/dbbench/tuxedo/c1.env"

*GROUPS

"group1"LMID="c1" GRPNO=1
"group2"LMID="c1" GRPNO=2
"group3"LMID="c1" GRPNO=3
"group4"LMID="c1" GRPNO=4
"group5"LMID="c1" GRPNO=5
"group6"LMID="c1" GRPNO=6
"group7"LMID="c1" GRPNO=7
"group8"LMID="c1" GRPNO=8
"group9"LMID="c1" GRPNO=9
"group10"LMID="c1" GRPNO=10
```

"group11"LMID="c1" GRPNO=11
"group12"LMID="c1" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"

*NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12

*ROUTING
"route1"FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="1-22:group1,23-44:group2,45-66:group3,67-88:group4,89-110:group5,111-132:group6,133-154:group7,155-176:group8,177-198:group9,199-220:group10,221-242:group11,243-264:group12,*:*"
::::::::::::
ubbconfig.c2
::::::::::::
#

```
#     ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY       40001   # IPC KEY from 32,768 to 16,777,215
MASTER       c2      # machine on which master copy is found
UID          30      # user id as displayed by command "id"
GID          5433    # group id as displayed by command "id"
PERM         0666    # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS   3700# max no of processes accesing bulleting board
MAXGTT       1000    # maximum simultaneous global transactions
MAXSERVERS   200     # maximum number of servers
MAXSERVICES  3100    # 991022 to big OK?.old=200 #maximum number of services
MAXCONV      1
MODEL        SHM     # SHM=single processor, MP=multi processor
LDBAL        N       # load balancing, Y=yes, N=no
CMTRET       COMPLETE
#MAXBUFTYPE   16      # maximum buffer types
#MAXBUFSTYPE  32      # maximum buffer subtypes
SCANUNIT     30      # scan program wake-up time in secs.
SANITYSCAN   5       # sanity scan wake-up
DBBLWAIT     1       # scanunit multiplier for DBBL max time wait
BBLQUERY     60      # check out wake-up time
BLOCKTIME    10      # blocking call time-out
NOTIFY       DIPIN
SYSTEM_ACCESS  FASTPATH
USIGNAL      SIGUSR2


*MACHINES
"c2"   LMID="c2"
     TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c2"
     ROOTDIR="/export/home/tuxedo"
     APPDIR="/export/home/dbbench/tuxedo"
     ULOGPFX="/export/home/dbbench/tuxedo/ULOGc2"
     ENVFILE="/export/home/dbbench/tuxedo/c2.env"

*GROUPS

"group1"LMID="c2" GRPNO=1
"group2"LMID="c2" GRPNO=2
"group3"LMID="c2" GRPNO=3
"group4"LMID="c2" GRPNO=4
"group5"LMID="c2" GRPNO=5
"group6"LMID="c2" GRPNO=6
"group7"LMID="c2" GRPNO=7
"group8"LMID="c2" GRPNO=8
"group9"LMID="c2" GRPNO=9
"group10"LMID="c2" GRPNO=10
"group11"LMID="c2" GRPNO=11
"group12"LMID="c2" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
```

TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"

*NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12

*ROUTING
"route1"FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="265-286:group1,287-308:group2,309-330:group3,331-352:group4,353-374:group5,375-
396:group6,397-418:group7,419-440:group8,441-462:group9,463-484:group10,485-506:group11,507-528:group12,*:*"
:::::::::::::::
ubbconfig.c3
:::::::::::::::
#
ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY   40001   # IPC KEY from 32,768 to 16,777,215
MASTER   c3   # machine on which master copy is found
UID   30   # user id as displayed by command "id"

```
GID          5433   # group id as displayed by command "id"
PERM         0666   # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS   3700# max no of processes accesing bulleting board
MAXGTT       1000   # maximum simultaneous global transactions
MAXSERVERS   200    # maximum number of servers
MAXSERVICES  3100   # 991022 to big OK?.old=200 #maximum number of services
MAXCONV      1
MODEL        SHM    # SHM=single processor, MP=multi processor
LDBAL        N      # load balancing, Y=yes, N=no
CMTRET       COMPLETE
#MAXBUFTYPE  16     # maximum buffer types
#MAXBUFSTYPE 32     # maximum buffer subtypes
SCANUNIT     30     # scan program wake-up time in secs.
SANITYSCAN   5      # sanity scan wake-up
DBBLWAIT     1      # scanunit multiplier for DBBL max time wait
BBLQUERY     60     # check out wake-up time
BLOCKTIME    10     # blocking call time-out
NOTIFY       DIPIN
SYSTEM_ACCESS  FASTPATH
USIGNAL      SIGUSR2


*MACHINES
"c3"   LMID="c3"
       TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c3"
       ROOTDIR="/export/home/tuxedo"
       APPDIR="/export/home/dbbench/tuxedo"
       ULOGPFX="/export/home/dbbench/tuxedo/ULOGc3"
       ENVFILE="/export/home/dbbench/tuxedo/c3.env"

*GROUPS

"group1"LMID="c3" GRPNO=1
"group2"LMID="c3" GRPNO=2
"group3"LMID="c3" GRPNO=3
"group4"LMID="c3" GRPNO=4
"group5"LMID="c3" GRPNO=5
"group6"LMID="c3" GRPNO=6
"group7"LMID="c3" GRPNO=7
"group8"LMID="c3" GRPNO=8
"group9"LMID="c3" GRPNO=9
"group10"LMID="c3" GRPNO=10
"group11"LMID="c3" GRPNO=11
"group12"LMID="c3" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
```

```
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"

# *NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12


*ROUTING
"route1"FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="529-550:group1,551-572:group2,573-594:group3,595-616:group4,617-638:group5,639-
660:group6,661-682:group7,683-704:group8,705-726:group9,727-748:group10,749-770:group11,771-792:group12,*:*"
:::::::::::::
ubbconfig.c4
:::::::::::::
#
#       ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY      40001   # IPC KEY from 32,768 to 16,777,215
MASTER      c4      # machine on which master copy is found
UID         30      # user id as displayed by command "id"
GID         5433    # group id as displayed by command "id"
PERM        0666    # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS   3700# max no of processes accesing bulleting board
MAXGTT      1000    # maximum simultaneous global transactions
MAXSERVERS    200   # maximum number of servers
MAXSERVICES   3100  # 991022 to big OK?.old=200 #maximum number of services
```

```
MAXCONV      1
MODEL        SHM    # SHM=single processor, MP=multi processor
LDBAL        N      # load balancing, Y=yes, N=no
CMTRET       COMPLETE
#MAXBUFTYPE  16     # maximum buffer types
#MAXBUFSTYPE 32     # maximum buffer subtypes
SCANUNIT     30     # scan program wake-up time in secs.
SANITYSCAN   5      # sanity scan wake-up
DBBLWAIT     1      # scanunit multiplier for DBBL max time wait
BBLQUERY     60     # check out wake-up time
BLOCKTIME    10     # blocking call time-out
NOTIFY       DIPIN
SYSTEM_ACCESS  FASTPATH
USIGNAL      SIGUSR2


*MACHINES
"c4"   LMID="c4"
     TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c4"
     ROOTDIR="/export/home/tuxedo"
     APPDIR="/export/home/dbbench/tuxedo"
     ULOGPFX="/export/home/dbbench/tuxedo/ULOGc4"
     ENVFILE="/export/home/dbbench/tuxedo/c4.env"

*GROUPS

"group1"LMID="c4" GRPNO=1
"group2"LMID="c4" GRPNO=2
"group3"LMID="c4" GRPNO=3
"group4"LMID="c4" GRPNO=4
"group5"LMID="c4" GRPNO=5
"group6"LMID="c4" GRPNO=6
"group7"LMID="c4" GRPNO=7
"group8"LMID="c4" GRPNO=8
"group9"LMID="c4" GRPNO=9
"group10"LMID="c4" GRPNO=10
"group11"LMID="c4" GRPNO=11
"group12"LMID="c4" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
```

```
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"

# *NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12


*ROUTING
"route1"FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="793-814:group1,815-836:group2,837-858:group3,859-880:group4,881-902:group5,903-
924:group6,925-946:group7,947-968:group8,969-990:group9,991-1012:group10,1013-1034:group11,1035-
1056:group12,*:*"
::::::::::::::
ubbconfig.c5
::::::::::::::
#
#       ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY          40001   # IPC KEY from 32,768 to 16,777,215
MASTER          c5      # machine on which master copy is found
UID             30      # user id as displayed by command "id"
GID             5433    # group id as displayed by command "id"
PERM            0666    # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS    3700# max no of processes accesing bulleting board
MAXGTT          1000    # maximum simultaneous global transactions
MAXSERVERS      200     # maximum number of servers
MAXSERVICES     3100    # 991022 to big OK?.old=200 #maximum number of services
MAXCONV         1
MODEL           SHM     # SHM=single processor, MP=multi processor
LDBAL           N       # load balancing, Y=yes, N=no
CMTRET          COMPLETE
#MAXBUFTYPE     16      # maximum buffer types
```

```
#MAXBUFSTYPE   32     # maximum buffer subtypes
SCANUNIT       30     # scan program wake-up time in secs.
SANITYSCAN      5     # sanity scan wake-up
DBBLWAIT        1     # scanunit multiplier for DBBL max time wait
BBLQUERY       60     # check out wake-up time
BLOCKTIME      10     # blocking call time-out
NOTIFY       DIPIN
SYSTEM_ACCESS  FASTPATH
USIGNAL       SIGUSR2


*MACHINES
"c5"   LMID="c5"
    TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c5"
    ROOTDIR="/export/home/tuxedo"
    APPDIR="/export/home/dbbench/tuxedo"
    ULOGPFX="/export/home/dbbench/tuxedo/ULOGc5"
    ENVFILE="/export/home/dbbench/tuxedo/c5.env"

*GROUPS

"group1"LMID="c5" GRPNO=1
"group2"LMID="c5" GRPNO=2
"group3"LMID="c5" GRPNO=3
"group4"LMID="c5" GRPNO=4
"group5"LMID="c5" GRPNO=5
"group6"LMID="c5" GRPNO=6
"group7"LMID="c5" GRPNO=7
"group8"LMID="c5" GRPNO=8
"group9"LMID="c5" GRPNO=9
"group10"LMID="c5" GRPNO=10
"group11"LMID="c5" GRPNO=11
"group12"LMID="c5" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
```

TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"

*NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12

*ROUTING
"route1"FIELD=FML_TERM
       BUFTYPE="FML"
       RANGES="1057-1078:group1,1079-1100:group2,1101-1122:group3,1123-1144:group4,1145-1166:group5,1167-
1188:group6,1189-1210:group7,1211-1232:group8,1233-1254:group9,1255-1276:group10,1277-1298:group11,1299-
1320:group12,*:*"
::::::::::::::
ubbconfig.c6
::::::::::::::
#
ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY   40001 # IPC KEY from 32,768 to 16,777,215
MASTER   c6   # machine on which master copy is found
UID   30   # user id as displayed by command "id"
GID   5433   # group id as displayed by command "id"
PERM   0666   # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS   3700# max no of processes accesing bulleting board
MAXGTT   1000   # maximum simultaneous global transactions
MAXSERVERS   200   # maximum number of servers
MAXSERVICES   3100   # 991022 to big OK?.old=200 #maximum number of services
MAXCONV   1
MODEL   SHM   # SHM=single processor, MP=multi processor
LDBAL   N   # load balancing, Y=yes, N=no
CMTRET   COMPLETE
#MAXBUFTYPE   16   # maximum buffer types
#MAXBUFSTYPE   32   # maximum buffer subtypes
SCANUNIT   30   # scan program wake-up time in secs.
SANITYSCAN   5   # sanity scan wake-up
DBBLWAIT   1   # scanunit multiplier for DBBL max time wait
BBLQUERY   60   # check out wake-up time
BLOCKTIME   10   # blocking call time-out

```
NOTIFY      DIPIN
SYSTEM_ACCESS   FASTPATH
USIGNAL     SIGUSR2


*MACHINES
"c6"   LMID="c6"
    TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c6"
    ROOTDIR="/export/home/tuxedo"
    APPDIR="/export/home/dbbench/tuxedo"
    ULOGPFX="/export/home/dbbench/tuxedo/ULOGc6"
    ENVFILE="/export/home/dbbench/tuxedo/c6.env"

*GROUPS

"group1"LMID="c6" GRPNO=1
"group2"LMID="c6" GRPNO=2
"group3"LMID="c6" GRPNO=3
"group4"LMID="c6" GRPNO=4
"group5"LMID="c6" GRPNO=5
"group6"LMID="c6" GRPNO=6
"group7"LMID="c6" GRPNO=7
"group8"LMID="c6" GRPNO=8
"group9"LMID="c6" GRPNO=9
"group10"LMID="c6" GRPNO=10
"group11"LMID="c6" GRPNO=11
"group12"LMID="c6" GRPNO=12


*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"

# *NETWORK
```

```
*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12


*ROUTING
"route1"FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="1321-1342:group1,1343-1364:group2,1365-1386:group3,1387-1408:group4,1409-1430:group5,1431-
1452:group6,1453-1474:group7,1475-1496:group8,1497-1518:group9,1519-1540:group10,1541-1562:group11,1563-
1584:group12,*:*"
::::::::::::::
ubbconfig.c7
::::::::::::::
#
#    ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY        40001   # IPC KEY from 32,768 to 16,777,215
MASTER        c7      # machine on which master copy is found
UID           30      # user id as displayed by command "id"
GID           5433    # group id as displayed by command "id"
PERM          0666    # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS  3700# max no of processes accesing bulleting board
MAXGTT        1000    # maximum simultaneous global transactions
MAXSERVERS    200     # maximum number of servers
MAXSERVICES   3100    # 991022 to big OK?.old=200 #maximum number of services
MAXCONV       1
MODEL         SHM     # SHM=single processor, MP=multi processor
LDBAL         N       # load balancing, Y=yes, N=no
CMTRET        COMPLETE
#MAXBUFTYPE   16      # maximum buffer types
#MAXBUFSTYPE  32      # maximum buffer subtypes
SCANUNIT      30      # scan program wake-up time in secs.
SANITYSCAN    5       # sanity scan wake-up
DBBLWAIT      1       # scanunit multiplier for DBBL max time wait
BBLQUERY      60      # check out wake-up time
BLOCKTIME     10      # blocking call time-out
NOTIFY        DIPIN
SYSTEM_ACCESS FASTPATH
USIGNAL       SIGUSR2


*MACHINES
```

"c7"   LMID="c7"
    TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c7"
    ROOTDIR="/export/home/tuxedo"
    APPDIR="/export/home/dbbench/tuxedo"
    ULOGPFX="/export/home/dbbench/tuxedo/ULOGc7"
    ENVFILE="/export/home/dbbench/tuxedo/c7.env"

*GROUPS

"group1"LMID="c7" GRPNO=1
"group2"LMID="c7" GRPNO=2
"group3"LMID="c7" GRPNO=3
"group4"LMID="c7" GRPNO=4
"group5"LMID="c7" GRPNO=5
"group6"LMID="c7" GRPNO=6
"group7"LMID="c7" GRPNO=7
"group8"LMID="c7" GRPNO=8
"group9"LMID="c7" GRPNO=9
"group10"LMID="c7" GRPNO=10
"group11"LMID="c7" GRPNO=11
"group12"LMID="c7" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env1"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"

*NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4

"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12

*ROUTING
"route1"FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="1585-1606:group1,1607-1628:group2,1629-1650:group3,1651-1672:group4,1673-1694:group5,1695-
1716:group6,1717-1738:group7,1739-1760:group8,1761-1782:group9,1783-1804:group10,1805-1826:group11,1827-
1848:group12,*:*"
:::::::::::::::
ubbconfig.c8
:::::::::::::::
#
ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY   40001   # IPC KEY from 32,768 to 16,777,215
MASTER   c8   # machine on which master copy is found
UID   30   # user id as displayed by command "id"
GID   5433   # group id as displayed by command "id"
PERM   0666   # UNIX permission from 0001 to 0777 in octal
MAXACCESSERS   3700# max no of processes accesing bulleting board
MAXGTT   1000   # maximum simultaneous global transactions
MAXSERVERS   200   # maximum number of servers
MAXSERVICES   3100   # 991022 to big OK?.old=200 #maximum number of services
MAXCONV   1
MODEL   SHM   # SHM=single processor, MP=multi processor
LDBAL   N   # load balancing, Y=yes, N=no
CMTRET   COMPLETE
#MAXBUFTYPE   16   # maximum buffer types
#MAXBUFSTYPE   32   # maximum buffer subtypes
SCANUNIT   30   # scan program wake-up time in secs.
SANITYSCAN   5   # sanity scan wake-up
DBBLWAIT   1   # scanunit multiplier for DBBL max time wait
BBLQUERY   60   # check out wake-up time
BLOCKTIME   10   # blocking call time-out
NOTIFY   DIPIN
SYSTEM_ACCESS   FASTPATH
USIGNAL   SIGUSR2

*MACHINES
"c8"   LMID="c8"
      TUXCONFIG="/export/home/dbbench/tuxedo/tuxconfig.c8"
      ROOTDIR="/export/home/tuxedo"
      APPDIR="/export/home/dbbench/tuxedo"
      ULOGPFX="/export/home/dbbench/tuxedo/ULOGc8"
      ENVFILE="/export/home/dbbench/tuxedo/c8.env"

*GROUPS

"group1"LMID="c8" GRPNO=1
"group2"LMID="c8" GRPNO=2
"group3"LMID="c8" GRPNO=3
"group4"LMID="c8" GRPNO=4
"group5"LMID="c8" GRPNO=5
"group6"LMID="c8" GRPNO=6
"group7"LMID="c8" GRPNO=7
"group8"LMID="c8" GRPNO=8
"group9"LMID="c8" GRPNO=9
"group10"LMID="c8" GRPNO=10
"group11"LMID="c8" GRPNO=11
"group12"LMID="c8" GRPNO=12

*SERVERS
DEFAULT:RESTART=Y MAXGEN=5 REPLYQ=N RQPERM=0660
TPCC SRVGRP=group1 RQADDR=TPCCq1 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group2 RQADDR=TPCCq2 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group3 RQADDR=TPCCq3 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group4 RQADDR=TPCCq4 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env2"
TPCC SRVGRP=group5 RQADDR=TPCCq5 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group6 RQADDR=TPCCq6 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group7 RQADDR=TPCCq7 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group8 RQADDR=TPCCq8 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group9 RQADDR=TPCCq9 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group10 RQADDR=TPCCq10 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group11 RQADDR=TPCCq11 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"
TPCC SRVGRP=group12 RQADDR=TPCCq12 SRVID=1 CLOPT="-s TPCC:TPCC"
ENVFILE="/export/home/dbbench/tuxedo/symfo.env3"

*NETWORK

*SERVICES
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group1
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group2
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group3
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group4
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group5
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group6
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group7
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group8
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group9
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group10

"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group11
"TPCC"TRANTIME=0ROUTING="route1" SRVGRP=group12

*ROUTING
"route1"FIELD=FML_TERM
      BUFTYPE="FML"
      RANGES="1849-1870:group1,1871-1892:group2,1893-1914:group3,1915-1936:group4,1937-1958:group5,1959-1980:group6,1981-2002:group7,2003-2024:group8,2025-2046:group9,2047-2068:group10,2069-2090:group11,2091-2112:group12,*:*"

Compilation Flags

These are the compilation flags used to compile the application code:

-O -lc -w -lintl

-O -lc -l./libOLINSERT.so -l/usr/lib/libc.so -l/usr/lib/libdl.so -l/opt/FCOBOL/lib -fast -xO4 -xspace -xarch=v8a -xchip=ultra

Appendix D: Disk Storage D≡

The calculations used to determine the storage requirements for the **8 hour logical log** and the **180-day** space calculations are contained in this appendix.

The calculations for the **8 hour recovery log** was determined as follows :

The number of logpages used during the measurement run was determined by using a stored procedure *sp_helpdb* tpcc before and after the run. We found the amount of log space used by the DBMS during the benchmark run. This was **15801328 KB.** The amount of log disk used per transaction was **15801328/4114487 = 3.84 KB.** Therefore we need **50268.07*60*8*3.84*1KB = 88.36 GB**

We allocated 48 * 9GB disks for the logs and the same for the mirrors.

180 day

Note : Numbers are in KBytes unless otherwise specified					
Warehouses	2,112.0	tpmC	25375.03	tpmC/W	12.01

Table	Rows	Data	Index	5% Space	8H Space	Total Space
Warehouse	2,112	2,137	0	107		2,244
District	21,120	21,168	0	1,058		22,226
Item	100,000	14,288	0	714		15,002
New–order	19,008,000	1,927,008	562,176	124,459		2,613,643
History	63,360,000	4,970,112	0		955,430	5,925,542
Orders	63,360,000	4,241,760	1,652,736		1,133,129	7,027,625
Customer	63,360,000	50,688,480	1,506,256	2,609,737		54,804,473
Order–line	633,605,888	68,336,672	0		13,136,705	81,473,377
Stock	211,200,000	76,800,240	0	3,840,012		80,640,252
DIRECTORY FILE		38,912				38,912
Dictionary		1,723,680				1,723,680
Totals		208,764,457	3,721,168	6,576,088	15,225,264	234,286,977

Table	Freespace
Warehouse	96,599
District	36,600
Item	715
New–order	923,904
History	1,258,752
Orders	1,195,392
Customer	2,534,400
Order–line	16,530,400
Stock	3,840,000
Indexes	2456624
Total	28,873,386

Dynamic space	77,548,544.00	Sum of Data for Order, Order–Line and History (excluding free extents)
Static space	141,513,168.65	Data + Index + 5% Space + Overhead – Dynamic space
Free space	22,297,298.35	
Daily growth	14,907,550.23	(Dynamic space/W * 62.5)* tpmC
Daily spread	0.00	Free–Space – 1.5 * Daily–Growth
180 day (KB)	2,824,872,210	2,824,872,210
180 day (GB)	2,694.01	2,694.01 <– Assumes no Daily Spread
Maximum	22,361,325.35	free space allowed

Measured Configuration			Space Usage	
Type	Number	Total GB	Usage	Size (GB)
9GB Drives	193	1,627.27	180–day Space	2,694.01
18GB Drives	11	185.51	Root,swap,usr	16.86
			Log	118.05
Totals	204	1,812.79	Total	2,828.92
			Deficit	1,016.13

Page 1

180 day

MB log used   6867
Total N–O Txn   1,255,446.000
KB log / Txn   5.6
8 Hour Log (GB)   64.54 of archive log
After image Log [GB]   5.86
Before image Log   1.95
Log index   0.10
Log disks[18GB]   4 8 Hour Log
Log disks [9GB]   6 (AI Log*mr)+(BI Log ^ mr)+(IX Log ^ mr)

Extra log space   37.69
Deficit after including excess log space   978.45
Replace 18GB from 9GB disks needed   **117.000**

Priced Configuration
                Type   Number   Total GB
        9GB Drives   76   640.79
        18GB Drives   128   2,158.71

Totals   204   2,799.50

Disk Capacities
                   9GB   18GB
                   512   512 bytes/sector
                   133   248 sectors/track
                   27   19 tracks/cylinder
                   3591   4712 sectors/cylinder
                   4926   7508 cylinders
                   4,924   7506 accessible cylinders
                   8.43   16.86
                   9gbsize   18gbsize
Non–DB space used
OS (root+etc)   12.86
Swap   4

Page 2

D

Appendix E: Driver Scripts

The following code sections show how the transactions are generated and how
statistics are gathered. Each of the transaction functions generates the input
data for that transaction, sends it to the client, reads the output form and
computes keying, response and think time statistics.

This is the main loop of the RTE:

```
/* run for ramp up without capturing the stats */
     i=0;
     in_ramp = 1;
     while (1)
     {
          tx_type = do_menu();/* Select transaction */
          switch (tx_type) {
          case NEWORDER:
                 do_neworder();
                 break;
          case PAYMENT:
                 do_payment();
                 break;
          case DELIVERY:
                 do_delivery();
                 break;
          case ORDSTAT:
                 do_ordstat();
                 break;
          case STOCKLEVEL:
                 do_stocklevel();
                 break;
          default:
                 fprintf(stderr, "%s: Slave %d: Internal error. Tx-type = %d\n",
                 hostname, slave_num, tx_type);
                 cleanup(-1);
          }
          end_time = gettime();
```

```
                if ( end_time >= control->end_rampup &&
                            end_time < control->end_stdystate )
                            in_ramp = 0;
                else
                        in_ramp = 1;
                if (end_time >= control->end_rampdown)
                        break;
        }
```
The do_menu function selects the transaction to execute based on the weighted distribution
algorithm.
```
int
do_menu()
{
        int val, result, menu_start, menu_end, menu_resp;
        char ch;
        /* Read menu line from client */
        /* Choose tx. type*/
        /* Now select menu and compute menu response time */
        menu_start = gettime();
        /* Write menu selection to client */
        /* Read input form for this transaction type */
        menu_end = gettime();
        menu_resp = menu_end - menu_start;
        if ( ! in_ramp) {
                statsp->menu_resp += menu_resp;
                /* Post in histogram bucket */
                if ((menu_resp / MENU_BUCKET) < MENU_MAX)
                        statsp->menu_hist[menu_resp / MENU_BUCKET]++;
                else
                        statsp->menu_hist[MENU_MAX - 1]++;
                if (menu_resp > statsp->menu_max)
                        statsp->menu_max = menu_resp;
        }
        return(result);
}
/*
 * Function: do_neworder
 * This function executes the neworder transaction
 * It generates all the input fields, sends it to the
 * client over the keying time, measures the response
 * time, reads the results and delays for the think time.
 */
 /* The code for the other transactions is similar */
do_neworder()
{
        struct newo_fld no;
        struct items_fld *itemp = no.items;
        int ol_cnt, rbk, remote = 0, i, x;
        char *bufp = fldbuf;
        int start_time, end_time, key_time, resp_time, elapse_time, del;
        start_time = gettime();
        /* Now wait for keying time */
        poll (0, 0, NEWO_KEY);
        /* Generate all input data */
        no.d_id = random(1, 10);
        no.c_id = NURand(1023, 1, 3000, CONST_CID);
```

```
ol_cnt = random(5, 15);
rbk = random(1, 100);/* trans. to be rolledback */
sprintf(bufp, "%02d%04d", no.d_id, no.c_id);
bufp += strlen(bufp);
/* Generate all the item fields */
for (i=0; i < ol_cnt; i++, itemp++) {
        itemp->ol_i_id = NURand(8191, 1, 100000, CONST_IID);
        /* If last item and rbk, select unused item */
        if (i == ol_cnt - 1 && rbk == 1) {
                itemp->ol_i_id = 100001;
        }
        x = random(1, 100);
        if ( x > 1)
                itemp->ol_supply_w_id = W_ID;
        else {
                /* Select a warehouse other than w_id */
                do {
                        x = random(1, control->scale);
                } while (x == W_ID);
                itemp->ol_supply_w_id = x;
                remote++;
        }
        itemp->ol_quantity = random(1, 10);
        sprintf(bufp, "%04d%06d%02d",itemp->ol_supply_w_id,
        itemp->ol_i_id, itemp->ol_quantity);
        bufp += strlen(bufp);
}
strcpy(bufp, leave_key);
bufp += 2;
/* Compute keying time info */
end_time = gettime();
key_time = end_time - start_time;
start_time = end_time;

/* Now send fields to client */
/* Read output screen from client */
end_time = gettime();
/* Store elapse time info for thruput */
elapse_time = end_time - control->start_time;
/* compute the how long it took to run the tx */
resp_time = end_time - start_time + control->newo_delta;
/* Wait think time */
del = delay(control->newo_think, 5*control->newo_think);
poll(0, 0, del + control->newo_delta);
end_time = gettime();
/* Now post all stats */
if ( ! in_ramp && end_time <= control->end_stdystate) {
        statsp->newo_cnt++; /* another one bytes the dust */
        if ( rbk == 1 )
                statsp->newo_rbkcnt++;
        statsp->newo_remote += remote;
        statsp->newo_olcnt += ol_cnt;
        statsp->newo_key += key_time;
        /* Save keying time in histogram bucket */
        statsp->newo_resp += (double) resp_time;/* sum up the response time */
        /* Save response time in histogram bucket */
```

```
                                statsp->newo_think += (double) del;
                                /* Save think time in histogram bucket */
                        }
                }
```

Appendix F: Screen Layouts

This Appendix contains the screen form layouts for the 5 transactions.

```
New-Order(N)  Payment(P)  Order-Status(O)  Delivery(D)  Stock-Level(S)  Exit(E)
                            New Order
Warehouse:        District: __                     Date:
Customer: ____    Name:                    Credit:     %Disc:
Order Number:           Number of Lines:          W_tax:        D_tax:

 Supp_W  Item_Id  Item Name                 Qty  Stock  B/G  Price    Amount
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
 ____    _____                              __
Execution Status:                                  Total:
                                                            **((
```

```
New-Order(N)  Payment(P)  Order-Status(O)  Delivery(D)  Stock-Level(S)  Exit(E)
                            Payment
Date:

Warehouse:                         District: __



Customer: ____  Cust-Warehouse: ____  Cust-District: __
Name:                    _____      Since:
                                              Credit:
                                              %Disc:
                                              Phone:

Amount Paid:         _____      New Cust-Balance:
Credit Limit:

Cust-Data:



                                                        **((
```

```
New-Order(N)  Payment(P)  Order-Status(O)  Delivery(D)  Stock-Level(S)  Exit(E)
                          Order-Status
Warehouse:        District: __
Customer: ____    Name:                    _____
Cust-Balance:

Order-Number:          Entry-Date:                      Carrier-Number:
Supply-W     Item-Id    Qty     Amount      Delivery-Date









                                                        **((
```

```
New-Order(N)  Payment(P)  Order-Status(O)  Delivery(D)  Stock-Level(S)  Exit(E)
                                    Delivery
Warehouse:

Carrier Number: __

Execution Status:




                                                                          **((
```

```
New-Order(N)  Payment(P)  Order-Status(O)  Delivery(D)  Stock-Level(S)  Exit(E)
                                  Stock-level
Warehouse:        District:

Stock level Threshold: __

Low Stock:




                                                                          **((
```

≡ *F*

Appendix G: Price Quotes   *G*≡

The following pages contain the pricing quotes for the hardware and software included in this FDR.

SALES QUOTE

August 21, 2000

131 Albright Way

Los Gatos, CA 95032-1801

Vinny Nguyen
Phone: 408-341-1743
Fax: 408-341-1696

	Quote #
	3262

Account Manager

Vinny Nguyen

Attention:

George Herman
Sun Micro Systems
901 San Antonio Rd.
Palo Alto, CA 94303

Phone: (650) 786-6271
Fax: (650) 786-6437

Ship To:

George Herman
Sun Micro Systems
901 San Antonio Rd.
Palo Alto, CA 94303

Phone: (650) 786-6271
Fax: (650) 786-6437
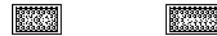
Here is the quote you requested. This quote is Valid for 60 Days.

Item	Mfg	Part	Qty	Description		Unit Price	Ext. Price
1	SUN	A25-CA	1	Sun Ent. 450 server in deskside tower; 4 CPU slots, 16 mem slots, 10 PCI I/O slots, 4 hot swap UltraSCSI disk bay, DVD, 1.44 floppy, removable media bay, 10/100 Ethernet, 2 Power supplies Solaris srvr license. No CPU, mem, or disk Rackmount		9,983.60	9,983.60
2	SUN	X2248A	4	480MHz UltraSPARC2 CPU 2/8MB E-cache and dc-dc converter for E450 Server		6,155.60	24,622.40
3	SUN	7005A	8	512-MB expansion memory (2x256 MB DIMMs) for E450		3,080.00	24,640.00
4	Sun	6601A	2	Factory installed 8 bay internal storage expansion kit for Enterprise 450 Server; includes 8 slot disk backplane, PCI UltraSCSI controller, and cables		998.80	1,997.60
5	SUN	5237A	21	18.2-GB/10000-rpm disk ULTRA SCSI disk drive 1-inch high		915.20	19,219.20
6	SUN	SG-XDSK020C-36G	2	StorEdge MultiPack with 36.4 GB (2 x 18.2 GB, 10000 rpm)		3,106.40	6,212.80
7	SUN	SG-XDSK060C-109G	30	Sun StorEdge MultiPack with 109.2 GB (6 x 18.2 GB, 10000 rpm)		6,089.60	182,688.00
8	SUN	6286A	1	12-24GB 4-mm DDS3 INT/E150		985.50	985.50
9	WYSE TECHNOLO GPT)	901237-01	1	WY-55 GREEN 80,132 COLUMN 14IN 2SERIAL WO/KEYBOARD		500.00	500.00

10	SUN	X3856A	16	68-pin to 68-pin 0.8 meter SCSI cable kit with power cord		39.60	633.60
11	SUN	530-2453	16	CBL,U SCSI-3/VHDCI,68P 2M		108.00	1,728.00
12	SUN	X311L	16	North American/Asian Power Cord			
13				SubTotal		31,962.30	273,210.70
14	SUN	A22UKC1Z9S-C512CP	8	Ultra 10 System, 440-MHz, 2-MB Ecache, PGx24 onboard graphics, 9 GB hard-disk drive, 32x CD-ROM, 512-MB DRAM, 1.44 Floppy preinstalled Solaris 7 Operating Environment; No Country Kit		4,527.60	36,220.80
15	SUN	X311L	8	North American/Asian Power Cord			
16	SUN	X7039A	8	OPT 512MB DRAM, 50NS, U10 only		1,579.60	12,636.80
17	SUN	1034A	8	QUAD FASTETHERNET CONTROLLER PCI ADAPTER		1,292.40	10,339.20
18	SUN	X7143A	8	17-Inch color Monitor, 15.7-Inch diagonal viewable, .28 dot pitch, 2-meter non detachable video, signal cable, DDC1/2B, MPR-II, TCO 95, CE Mark, Energy Star Compliant, 80W power consumption, 1152x900-66Hz, 76Hz, 1024x768-60Hz		286.00	2,288.00
19				SubTotal		7,685.60	61,484.80

Subtotal		$334,695.50
Sales Tax @	8.25%	27,612.38
		$362,307.88

Please contact me if I can be of further assistance.

This Proposal is a copyright of CAT Technology, Inc. and represents Systems Integration efforts not to be forwarded in whole or in part to third parties without the written consent of CAT Technology, Inc. ACCEPTED BY- X_____ Date:_____	Purchase Order Number: # _____ Remit To Address: CAT TECHNOLOGY PO Box 45124 San Francisco, CA 94145

Software House International
Pricing Proposal

Quotation #MO-200818-63735
08/18/2000

Sun Microsystems
George Herman
Quote Good for Ninety Days

Phone: Fax: 650-786-7353

SHI Account Exec: Matthew O. Martin
Telephone : (408) 922-1106
Fax :   (408) 526-1222

Reference:

Product	Part #	Qty	List	Your Price	Total
6 Port 10Bt Generic Hub + 1	Z222666	2904		$23.00	$66,792.00
8 Port 10/100 Hub	Z222667	3		$114.00	$342.00
Five yr return to man war.					
Total					**$67,134.00**

Additional Comments

THE ECOMMERCE TRANSACTION PLATFORM

bea

August 16, 2000

Mr. George Herman
TPC-C Performance Project Manager
Sun Microsystems
650-786-7353 FAX
650-786-6271

Dear Mr. Herman:

Per your request I am enclosing the pricing information regarding TUXEDO that you
requested. This pricing applies to Tuxedo 6.1, 6.2, 6.3, 6.4,6.5, and 7.1. Please note
that Tuxedo 7.1 is our most recent version of Tuxedo but that all 6.x releases are
generally available. Core functionality services pricing is appropriate for your activities.
As per the table below Sun Ultra Sparc systems are classified as either a Tier 1, 2, 3, 4
or 5 systems depending on the performance and CPU capacity of the system. This
quote is valid for 90 days from the date of this letter.

Tuxedo Core Functionality Services (CFS) Program Product Pricing and Description

TUX-CFS provides a basic level of middleware support for distributed computing, and
is best used by organizations with substantial resources and knowledge for advanced
distributed computing implementations.

TUX-CFS prices are server only and are based on the overall performance
characteristics of the server and uses the same five-tier computer classification as
TUXEDO 6.x and 7.1. Prices range from $3,000 for Tier 1 to $250,000 for Tier 5.
Under this pricing option EVERY system running TUX-CFS at the user site must have
a TUXEDO license installed and pay the appropriate per server license fees.
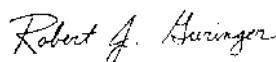
Very Truly Yours,

Robert J. Gieringer

Robert J. Gieringer
Worldwide Pricing Manager

BEA Tux/CFS Unlimited User License Fees Per Server

1

Unlimited User License fees per server	Number of Users	Dollar Amount	Maintenance (5 x 8) per year	Maintenance (7 x 24) per year
Tier 1 –– PC Servers with 1 or 2 CPUs, entry level RISC Uni–processor workstations and servers	Unlimited	$3,000.00	$480.00	$690.00
Tier 2 – PC Servers with 3 or 4 CPUs, Midrange RISC Uni–processor servers and workstations	Unlimited	$12,000.00	$1,920.00	$2,760.00
Tier 3 – Midrange Multiprocessors, up to 8 CPUs per system capacity	Unlimited	$30,000.00	$4,800.00	$6,900.00
Tier 4 – Large (more than 8, less than 32 CPUs)	Unlimited	$100,000.00	$16,000.00	$23,000.00
Tier 5 – Massively Parallel Systems, > 32 processors	Unlimited	$250,000.00	$40,000.00	$57,500.00

	Tier 1	Tier 2	Tier 3	Tier 3	Tier 4	Tier 5
Platform						
SUN SPARC	X– term–inal 1 Station 5/85

Station 4 Station 20/50 Station 20/51 Station 20/61 Ultra 1 140/170 Server 470 Server 5/70 Server 20/50 Server 20/51 Server 20/61 Ultra 2 Desktop Ultra 5, 5S Ultra 10, 10S Enterprise 1 &150 | Server 5/85 Station 20/71 Server 20/71 Enterprise 220R Enterprise 250 Enterprise 2 –2100, 2200 Ultra 60 | Station 20/502 MP 20/612 MP 20/514 MP 20/HS11 20/712 MP Server 1000 Server 1000E Server 20/502 Server 20/712 Server 20/612 Server 20/514 Enterprise 2 –2300 Enterprise 420R | Enterprise 450 SparcCenter 1000 Enterprise 3000 & 3500 Enterprise 4000, 4500, 5000 & 5500 < 8 proc. | SparcCenter 2000 SparcCenter 2000E Enterprise 4000, 4500, 5000 & 5500 Between 8 and 32 proc. Enterprise 6000 & 6500 Between 8 and 32 proc. CRS6400 (< 32 proc.) | CRS6400 (≥32 proc.) Enterprise 6000 & 6500 (≥32 proc.) Enterprise 10000 |

IntraServer

August 8, 2000

Mr. George Herman
Sun Microsystems, Inc. MS MPK12-112
901 San Antonio Road
Palo Alto, CA 94303-4900

Dear George,

Thank you for your interest in IntraServer's Host Adapter products. I am pleased to provide you this quote for ITIpof™ Host Adapters.

Quotation Number: Sun000000
Expiration Date: 10/19000

Part Number Description		Any! Quantity
ITI-6200U2-S	Dual port Ultra2/LVD SCSI PCI adapter	$462.50

*Available direct from IntraServer Technology, Inc.

The following general conditions apply to this quote:

- IntraServer products are manufactured in Massachusetts, USA.
- IntraServer products are offered with a return-to-factory limited warranty.
- IntraServer products offer technical support free of charge.

This quote is valid until the expiration date above. IntraServer products are FOB Holliston, MA USA.

Tim Collins - Business Development
IntraServer Technology, Inc.
7 October Hill Road
Holliston, MA 01746
Phone: 508-429-0425 ext. 233
Fax: 508-429-0430
http://www.intraserver.com

IntraServer Technology, Inc.
+1 508-429-0425

Seven October Hill Road
+1 508-429-0430 FAX

Holliston, MA 01746 USA

≡ G

TPC Benchmark C™ Full Disclosure—August 2000