HP 9000 V2500 Enterprise Server
using
HP-UX 11.0 64-bit
and
Oracle8i Enterprise Database Server v8.1.5

TPC Benchmark® C
Full Disclosure Report

Third Edition
Submitted for Review
August 6, 1999

Third Edition - August 6, 1999

Hewlett-Packard Company believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Hewlett-Packard Company assumes no responsibility for any errors that may appear in this document.
The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Hewlett-Packard Company provides no warranty of the pricing information in this document.
Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC Benchmark® C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.
All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Hewlett-Packard Company does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC®) or normalized price/performance ($/tpmC®). No warranty of system performance or price/performance is expressed or implied in this report.

Printed in U.S.A., August 6, 1999.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark® C test conducted on the HP 9000 V2500 Enterprise Server in a client/server configuration, using Oracle8i Enterprise Database Server v8.1.5 and the TUXEDO 6.4 transaction monitor. The operating system used for the benchmark was Hewlett-Packard's HP-UX 11.0 64-bit. The application was written in C and compiled using HP C/ANSI C/HP-UX.

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.

Standard and Executive Summary Statements

Page *iii* contains the standard system summary and pages ***iv-vi*** contain the executive summary of the benchmark results for the HP 9000 V2500 Enterprise Server.

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the price/performance, were audited by Tom Sawyer for Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

Standard System Summary

Company Name	**System Name**	**Database Software**	**Operating System Software**
Hewlett-Packard Company	HP 9000 V2500 Enterprise Server	Oracle8i Enterprise Database Server v8.1.5	HP-UX 11.0 64-bit Extension Pack 9903
HP H/W Availability Date - currently available S/W Availability Date - August 31, 1999			
Total System Cost	**TPC-C® Throughput**	**Price/Performance**	
Hardware Software 5-year maintenance	Sustained maximum throughput of System running TPC-C® expressed in transactions per minute	Total system cost/tpmC ($8,142,782/92832.96)	
$8,142,782	**92,832.96 tpmC**	**$87.71 per tpmC**	

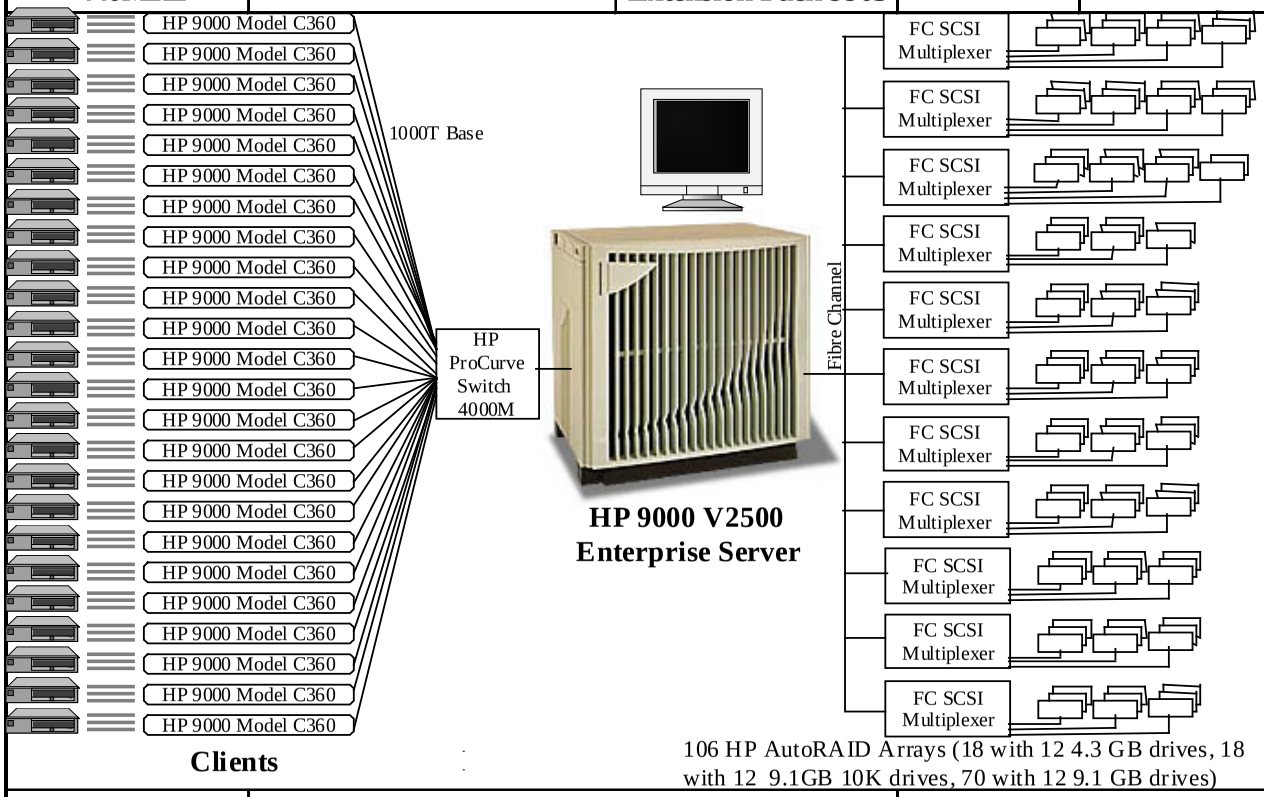
HP 9000 V2500 Enterprise Server
Client/Server with 24 C360 front-ends

TPC-C Revision 3.4

Report Date: August 6, 1999

Total System Cost	TPC Throughput	Price/Performance	Availability Date
$8,142,782	**92,832.96 tpmC**	**$87.71/tpmC**	**August 31, 1999**

Processors	Database Manager	Operating System	Other Software	Number of Users
32 PA-RISC 8500 440MHz	**Oracle8i Enterprise Database Server v8.1.5**	**HP-UX 11.0 64-bit Extension Pack 9903**	**TUXEDO 6.4**	**74,880**



System Components	*Server (HP 9000 V2500 Enterprise Server)*		*Each Client (24 C360)*	
	Qty	*Type*	*Qty*	*Type*
Processors	32	440MHz PA-RISC 8500	1	367MHz PA-RISC 8500
Cache Memory	each	0.5 MB I-cache, 1 MB D-cache	each	0.5 MB I-cache/1 MB D-cache
Memory	32	GB	1	1.5 GB
Disk Controllers	11	HP-PCI Fibre Channel	1	Ultra Wide SCSI
	11	FC SCSI Multiplexer		
	36	16 Bit FW SCSI Adapter		
Disk Drives	106	HP AutoRaid Arrays with 18 12-4.3 GB drives, 18 12-9.1 GB 10k disks and 70 12-9.1 GB drives	1	4 GB disk
Total Storage	8,102	GB		
Tape Drives	1	DDS Storage System		
Terminals	1	Console Terminal	1	Console Terminal

HEWLETT PACKARD / ORACLE	**HP 9000 V2500 Enterprise Server**	TPC-C Rev 3.4 Report Date: August 6, 1999

Description	Part Number	Brand	Price Key	Unit Price	Qty	5 Year Price	Maint. Price
Server Hardware							
HP 9000 V2500 Enterprise Server	A5074A		1	138,600	1	138,600	119,532
Add'l Dual 440 MHz PA-RISC 8500 CPUs	A5492A Opt. 0D1		1	45,500	16	728,000	118,272
System Mgmt. Station	A4802B		1	8,036	1	8,036	2,117
Memory Controller Board	A5078A, Opt. 0D1		1	7,000	4	28,000	
256MB DIMM for a total of 2 GB	A5082A, Opt. 0D1		1	24,500	16	392,000	
PCI Fibre Channel Adapter	A3740A, Opt. 0D1		1	1,890	11	20,790	
PCI 1000BT LAN Adapter	A4926A Opt 0D1		1	1,467	1	1,467	
4GB DDS-2 DAT Drive	A3183A, Opt 0DZ		1	1,156	1	1,156	
PCI 64-port Mux card	J3593A Opt. 0D1		1	1,047	1	1,047	
16 port RS232 DB25 port module	J2485A		1	616	1	616	
25 ft cable	J3595A		1	105	1	105	
5.5 kVA UPS	A3589A Opt. 002		1	7,000	1	7,000	336
1.8kVA HP UPS Rackmount	A2997A Opt. 002		1	2,310	53	122,430	16,027
2.0m Field Integrated Cabinet	C2787A		1	1,400	31	43,400	
FC SCSI Multiplexer	A3511A		1	7,280	11	80,080	71,019
16 Bit FW SCSI Adapter	A3511A, Opt. 003		1	907	36	32,634	
Auto Raid Array Model 12H	A3700AZ		1	3,050	106	323,300	560,210
Two 96 MB controllers with Auto Raid	Opt. 203		1	9,200	106	975,200	
Twelve 9.1 GB disk modules, 10K rpm			1	15,120	18	272,160	
Twelve 4.3 GB disk modules	Option 112		1	10,080	18	181,440	
Twelve 9.1 GB disk modules	Option 132		1	13,860	70	970,200	
0.9m 68-pin high density male connec.	Option 801		1	84	70	5,880	
2.5m HDTS68 to HDTS68 Cable	C2924A		1	98	36	3,528	
				Subtotal		**4,337,069**	**887,513**
Server Software							
Oracle8i Enterprise Database Server v8.1.5		Oracle	2		1	975,840	975,840
HP-UX 11.0 Sys Media, CD-ROM	B3920EA, Opt AAF		1	364	1	364	
				Subtotal		**976,204**	**975,840**
Client Hardware							
Hewlett Packard Model C360 Workstation	A4988A		1	9,100	24	218,400	75,802
256 MB ECC memory module	A4517A		1	1432	144	206,136	
Console	C1064GX		1	377	1	377	168
4 GB ultra SCSI disk	A4569A		1	700	24	16,800	
100 Base T PCI LAN Adapter	B5509BA		1	242	72	17,424	
				Subtotal		**459,137**	**75,970**
Client Software							
HP C/ANSI C Compiler	B3901BA, Option AH0		1	1,103	1	1,103	980
BEA Tuxedo 6.4		Bea Sys.	3	3,000	24	72,000	54,000
				Subtotal		**73,103**	**54,980**
User Connectivity							
(8+1) port 10Mbps Ethernet Hub, 913 spares	DEH2924	SW House	4	28.95	10,296	298,069	
HP ProCurve Switch 4000M	J4121A		1	2,771	1	2,771	588
HP ProCurve Switch Gigabit-SX Module	J4113A		1	1,049	1	1,049	
HP ProCurve Switch 10/100Base-T Module	J4111A		1	489	1	489	
				Subtotal		**302,378**	**588**
				Other discounts*		0	**$0**
				Total		**6,147,891**	**1,994,891**

Notes:1=Sun Data, 2=Oracle
3=BEA Systems, 4=Software House International

Audited by Tom Sawyer for Performance Metrics, Inc.

Five Year Cost of Ownership:	**$8,142,782**
tpmC Rating:	**92,832.96**
$/tpmC:	**$87.71**

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

Numerical Quantities Summary for HP 9000 V2500 Enterprise Server

MQTH, Computed Maximum Qualified Throughput **92,832.96 tpmC**

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	1.14s	4.55s	0.67s
Payment	1.02s	4.41s	0.56s
Order-Status	1.09s	3.54s	0.62s
Delivery (interactive portion)	0.08s	0.12s	0.07s
Delivery (deferred portion)	1.19s	4.33s	0.72s
Stock-Level	1.38s	4.43s	0.79s
Menu	0.10s	0.53s	0.001s

Transaction Mix, in percent of total transactions

New-Order	44.79%
Payment	43.06%
Order-Status	4.05%
Delivery	4.05%
Stock-Level	4.05%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18.02s	18.03s	18.04s	0.01s	12.13s	191.14s
Payment	3.01s	3.02s	3.03s	0.01s	12.06s	173.28s
Order-Status	2.01s	2.02s	2.03s	0.01s	10.06s	119.67s
Delivery (interactive)	2.01s	2.02s	2.03s	0.02s	5.07s	61.74s
Stock-Level	2.01s	2.02s	2.02s	0.01s	5.07s	53.89s

Test Duration

Ramp up time	46.75 minutes
Measurement interval	26 minutes
Transactions during measurement interval	5,388,722
Ramp down time	6.87 minutes

Checkpointing

Number of checkpoints in measurement interval	1
Checkpoint Interval	26 minutes

Reproducibility Run

Throughput	92,777.38 tpmC
Relative to MQTH	-0.06%

Preface

TPC Benchmark C Overview

This is the full disclosure report for a benchmark test of the HP 9000 V2500 Enterprise Server using Oracle8i Enterprise Database Server v8.1.5. It meets the requirements of the TPC Benchmark® C Standard Specification, Revision 3.4 dated August 25, 1998.

TPC Benchmark® C was developed by the **T**ransaction Processing **P**erformance **C**ouncil (TPC). It is the intent of this group to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Hewlett-Packard Company Oracle Corporation are active participants in the TPC.

*TPC Benchmark® C is an **O**n **L**ine **T**ransaction **P**rocessing (OLTP) workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:*

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

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

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C® approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to other environments are not recommended.

Hewlett-Packard Company does not warrant or represent that a user can or will achieve performance similar to the benchmark results contained in this report. No warranty of system performance or price/performance is expressed or implied by 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 output functions.

Appendix A contains the HP C/ANSI C/HP-UX application code used in this TPC-C® test.

1.2 Test Sponsor

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

The High Performance Systems Division of Hewlett-Packard Company and Oracle Corporation are the test sponsors of this TPC Benchmark® C.

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 options
- Recover/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.

The intent of the above clause is that anyone attempting to recreate the benchmark environment has sufficient information to compile, link, optimize, and execute all software used to produce the disclosed benchmark result.

Appendix A contains the application "make" files. Appendix C contains the HP-UX operating system parameters used to generate the kernel for the configuration used in this benchmark. Also included are all of the Oracle8i Enterprise Database Server v8.1.5 database parameters and the TUXEDO 6.4 transaction monitor parameters used.

1.4 Configuration Diagrams

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

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

The server System Under Test, an HP 9000 V2500 Enterprise Server depicted in Figure 1.1, consisted of:

- 32 440MHz PA-RISC 8500 System Processors

- 32 GB of memory
- 11 HP-PCI Fibre Channel Adapters, 11 FC SCSI Multiplexers, and 36 16 Bit DW SCSI Adapters
- 106 HP AutoRaid Arrays (with 36 12-4.3 GB disks, 18 12-9.1 GB 10k disks and 52 12-9.1 GB disks
- One LAN interfaces

As indicated in Figure 1.1, this benchmark configuration used Remote Terminal Emulator (RTE) programs that executed on 12 K460 Enterprise Server drivers to emulate TPC-C user sessions. The emulated users on the driver systems were connected through the same switch that connected the client systems to the system under test. Connections to the driver systems used 100 Base-T local area network (LAN) and communicated using TCP/IP. The clients were connected to the SUT via one HP ProCurve Switch 4000M switch.

The priced configuration for the HP 9000 V2500 Enterprise Server is shown in Figure 1.2. In the priced configuration, the RTE shown in the benchmark configuration is replaced by the appropriate number of workstations (emulating ANSI terminals) connected to hubs.

Figure 1.1: HP 9000 V2500 Enterprise Server Benchmark Configuration

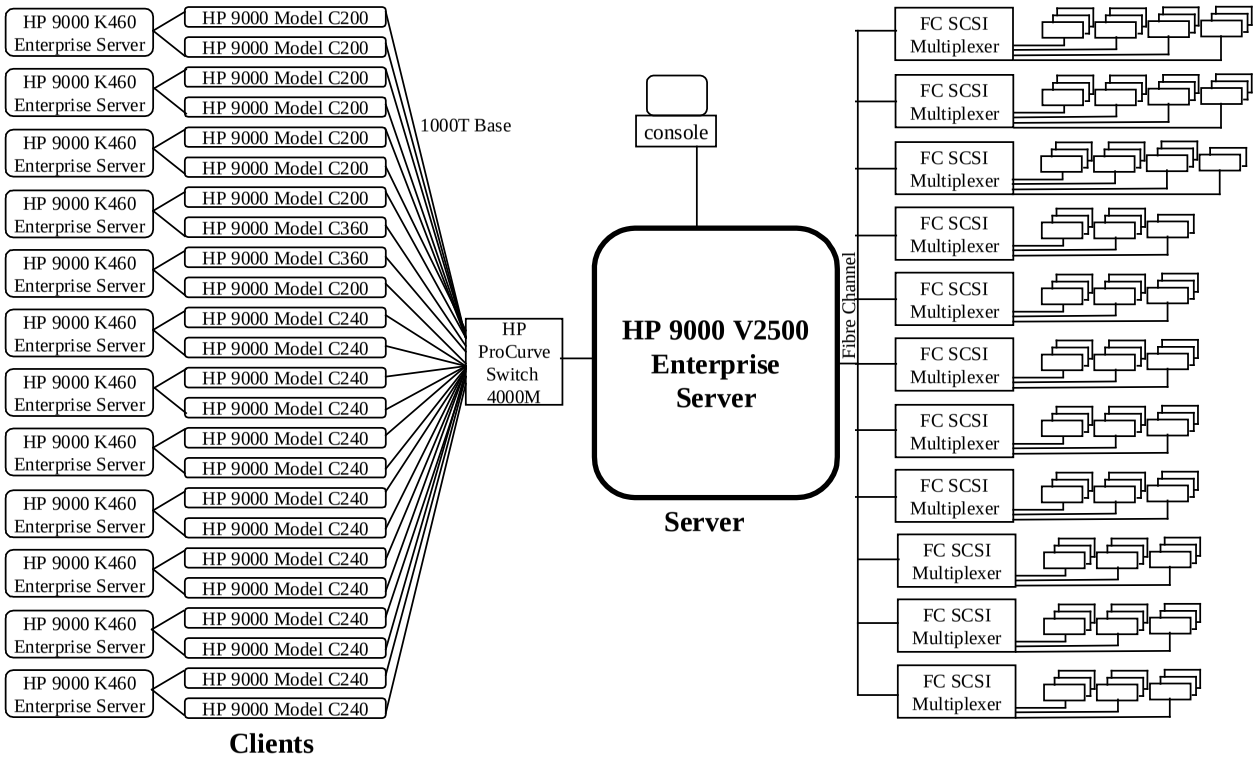
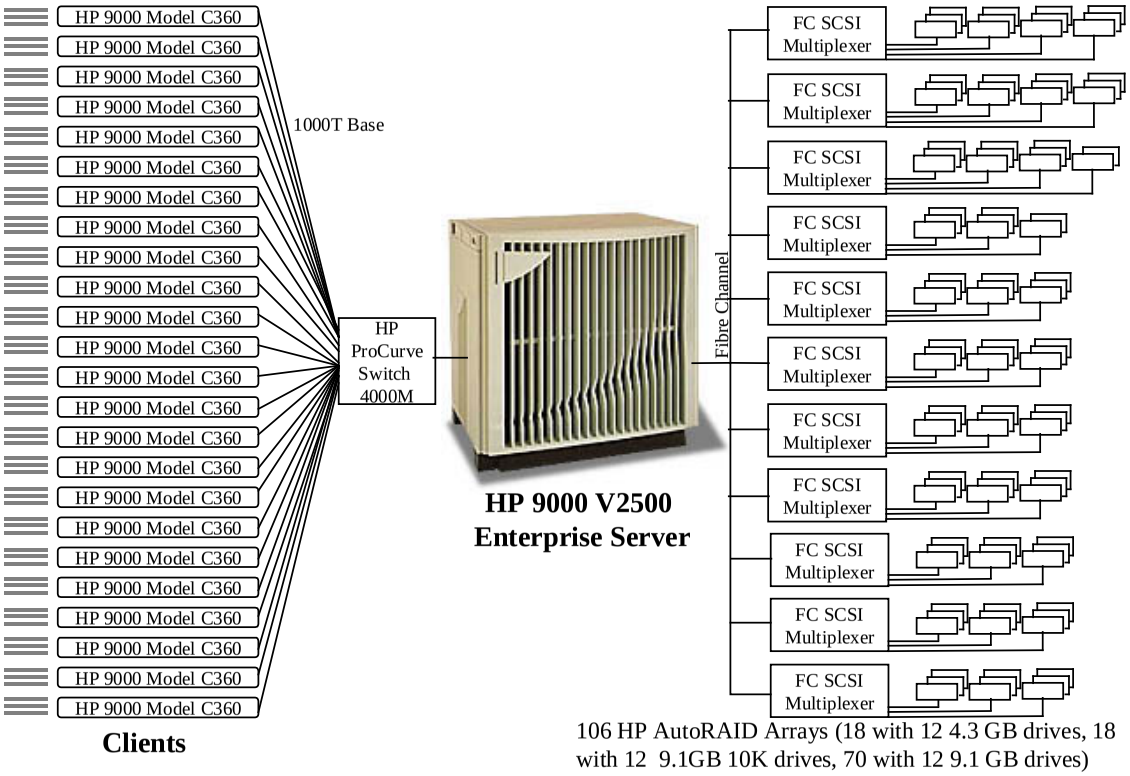


Figure 1.2: HP 9000 V2500 Enterprise Server Priced Configuration



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 the Oracle8i Enterprise Database Server v8.1.5 database for TPC-C® testing.

2.2 Physical Organization of Database

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

Space was allocated to Oracle8i Enterprise Database Server v8.1.5 according to the data in section 5.2. The size of the database table space on each disk drive was calculated to provide even distribution of load across the disk drives.

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.

There were no restrictions on insert and delete operations to any tables.

2.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C® benchmark, any such partitioning must be disclosed. Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

Partitioning, replication, and additional or duplicated attributes were not used in this implementation.

3 Clause 2 Related Items

3.1 Random Number Generation

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

The library routine SRAND48 (3C) was used to seed the library routine DRAND48 (3C) which generated pseudo-random numbers using the well-known linear congruential algorithm and 48-bit integer arithmetic. Further information on SRAND48 (3C) and DRAND48 (3C) can be found in the HP-UX Reference Manual Vol. 3.

3.2 Input/Output Screen Layout

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

The screen layouts corresponded exactly to those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C® Standard Specification.

3.3 Priced 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. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal features were verified by manually exercising each specification on an HP 712/80 workstation running an ANSI terminal emulator.

3.4 Presentation Manager or Intelligent Terminal

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

Application code running on the client implemented the TPC-C user interface. A listing of this code is included in Appendix A. Used capabilities of the terminal beyond basic ASCII entry and display were restricted to cursor positioning.

A presentation manager was not used.

Table 3.1: Transaction Statistics

Type	Item	Value
New Order	Home warehouse items	99.00%
	Remote warehouse items	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.01%
	Remote warehouse	14.99%
	Non primary key access	59.99%
Order Status	Non primary key access	59.95%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.79%
	Payment	43.06%
	Order Status	4.05%
	Delivery	4.05%
	Stock Level	4.05%

3.5 Transaction Statistics

Table 3.1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 require.

3.6 Queuing Mechanism

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

Delivery transactions were submitted to servers using the same TUXEDO mechanism that other transactions used. The only difference was that the call was asynchronous, i.e., control would return to the client process immediately and the deferred delivery part would complete 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 quotes the specification definition of each of these properties and describes the tests done as specified and monitored by the auditor to demonstrate compliance.

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, WAREHOUSE, and DISTRICT tables have been changed appropriately.

The values of w_ytd, d_ytd, c_balance, c_ytd_payment, and c_payment_cnt of a randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was committed. The values w_ytd, d_ytd, c_balance, c_ytd_payment, and c_payment_cnt were retrieved again. It was verified that all values had been changed appropriately.

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, WAREHOUSE, and DISTRICT tables have NOT been changed

The values of w_ytd, d_ytd, c_balance, c_ytd_payment and c_payment_cnt of a randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was rolled back. The values of w_ytd, d_ytd, c_balance, c_ytd_payment, c_payment_cnt were retrieved again. It was verified that none of the values had changed.

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 the database is initially in a consistent state.

The TPC Benchmark C standard requires the System Under Test to meet the following 12 consistency conditions (c.f. *TPC Standard Specification, Clauses 3.3.2.1 to 3.3.2.12*):

1. the sum of the district balances in a warehouse is equal to the warehouse balance;
2. for each district, the next order-id minus one is equal to maximum order-id in the ORDER table and equal to the maximum new-order-id in the NEW-ORDER table;
3. 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;

4. for each district, the sum of the order-line counts equals the number of rows in the ORDER-LINE table for that district;
5. for each row in the ORDER table, the carrier-id is set to a null value only if there is a corresponding row in the NEW-ORDER table;
6. for each row in the ORDER table, the order-line count must equal the number of rows in the ORDER-LINE table for that order;
7. for any row in the ORDER-LINE table, the delivery date/time is set to a null value only if the corresponding row in the ORDER table has the carrier-id set to a null value;
8. for each warehouse, the year-to-date amount must equal the sum of the amounts in the HISTORY table for that warehouse;
9. for each district, the year-to-date amount must equal the sum of the amounts in the HISTORY table for that district;
10. for each customer, the balance must equal the sum of the order-line amount minus the sum of the history amount for that customer;
11. for each district, the total orders minus the total new-orders must equal the sum of the customer delivery count;
12. for any randomly selected customer, the balance plus the year-to-date payment must equal the sum of the order-line amount.

The TPC Benchmark C Standard Specification requires explicit demonstration that the conditions are satisfied for the first four conditions only.

To demonstrate that consistency is maintained, conditions 1-4 were verified for a sample of warehouses before and after the durability tests.

4.4 Isolation

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

*This property is commonly called **serializability**. Sufficient conditions must be enabled at either the system or application level to ensure serializability of transactions under any arbitrary mix of TPC-C transactions, unless otherwise specified by the transaction profile. The system or application must have full serializability enabled (i.e., repeated reads of the same rows within any committed transaction must return identical data when run concurrently with any arbitrary mix of TPC-C transactions), except in the case of Stock-Level transaction. For the Stock-Level transaction, the isolation requirement is relaxed to simply require that the transaction see only committed data.*

The TPC Benchmark C Standard (Revision 3) defines nine required tests to be performed to demonstrate that the required levels of transaction isolation are met.

For conventional locking schemes, isolation should be tested as described below. Systems that implement other isolation schemes may require different validation techniques. It is the responsibility of the test sponsor to disclose those techniques and the tests for them. If isolation schemes other than conventional locking are used, it is permissible to implement these tests differently provided full details are disclosed. (Examples of different validation techniques are shown in Isolation Test 7, Clause 3.4.2.7).

4.4.1 Isolation Test 1

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T0 was executed for a randomly selected customer, and the order returned was noted. T0 was committed
2. A New-Order transaction T1 was started for the same customer used in T0. T1 was stopped prior to COMMIT.
3. An Order-Status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.
4. T1 was allowed to complete and was committed.
5. An Order-Status transaction T3 was started for the same customer used in T1. T3 returned the order inserted by T1.

This outcome demonstrates serialization of T2 before T1. It has equivalent validity to the outcome specified in the Standard which supposes T1 to be serialized before T2.

4.4.2 Isolation Test 2

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions when the New-Order transaction is ROLLED BACK.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T0 was executed for a randomly selected customer and the order returned was noted. T0 was committed.
2. A New-Order transaction T1 with an invalid item number, was started for the same customer used in T0. T1 was stopped immediately prior to ROLLBACK.
3. An Order-Status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.
4. T1 was allowed to ROLLBACK.
5. An Order-Status transaction T3 was started for the same customer used in T1. T3 returned the same order that T0 had returned.

4.4.3 Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions.

The execution of the above test proceeded as follows:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1 was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to COMMIT.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to complete. T2 completed and was committed.
5. The order number returned by T1 was the same as the D_NEXT_O_ID retrieved in step 1. The order number returned by T2 was one greater than the order number returned by T1.

6. The D_NEXT_O_ID of the same district was retrieved again. It had been incremented by two (i.e. it was one greater than the order number returned by T2).

4.4.4 Isolation Test 4

This test demonstrates isolation for write-write conflicts of two New-Order transactions when one transaction is ROLLED BACK.

The execution of the above test proceeded as follows:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1, with an invalid item number, was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to ROLLBACK.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to roll back, and T2 completed and was committed.
5. The order number returned by T2 was the same as the D_NEXT_O_ID retrieved in step 1.
6. The D_NEXT_O_ID of the same district was retrieved again. It had been incremented by one (i.e. one greater than the order number returned by T2).

4.4.5 Isolation Test 5

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions.

The execution of the above test proceeded as follows:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_BALANCE of the customer found in step 1 was retrieved.
3. A Delivery business transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the COMMIT of the database transaction corresponding to the district used in step 1.
4. A Payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to complete. T2 completed and was committed.
6. The C_BALANCE of the customer found in step 1 was retrieved again. The C_BALANCE reflected the results of both T1 and T2.

4.4.6 Isolation Test 6

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when the Delivery transaction is ROLLED BACK.

The execution of the above test proceeded as follows:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_BALANCE of the customer found in step 1 was retrieved.

3. A Delivery business transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the ROLLBACK of the database transaction corresponding to the district used in step 1.
4. A Payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to ROLLBACK. T2 completed and was committed.

The C_BALANCE of the customer found in step 1 was retrieved again. The C_BALANCE reflected the results of only T2.

4.4.7 Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the price of an item.

The execution of the above test proceeded as follows:

1. The I_PRICE of two randomly selected items X and Y were retrieved.
2. A New-Order transaction T2 with a group of items including items X and Y was started. T2 was stopped immediately after retrieving the prices of all items. The prices of items X and Y retrieved matched those retrieved in step 1.
3. A transaction T3 was started to increase the price of items X and Y by 10%.
4. T3 did not stall and no transaction was rolled back. T3 was committed.
5. T2 was resumed, and the prices of all items were retrieved again within T2. The prices of items X and Y matched those retrieved in step 1.
6. T2 was committed.
7. The prices of items X and Y were retrieved again. The values matched the values set by T3.

- Execution followed *Case D* of *Clause 3.4.2.7*.

4.4.8 Isolation Test 8

This test demonstrates isolation for phantom protection between New-Order and Order-Status transactions.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T1 was started for a randomly selected customer.
2. T1 was stopped immediately after reading the order table for the selected customer. The most recent order for that customer was found.
3. A New-Order transaction T2 was started for the same customer. T2 completed and was committed without being blocked by T1.
4. T1 was resumed and the ORDER table was read again to determine the most recent order for the same customer. The order found was the same as the one found in step 2.
5. T1 completed and was committed.

4.4.9 Isolation Test 9

This test demonstrates isolation for phantom protection between New-Order and Delivery transactions.

The execution of the above test proceeded as follows:

1. The NO_D_ID of all new_ORDER rows for a randomly selected warehouse and district was changed to 11. The changes were committed.
2. A Delivery transaction T1 was started for the selected warehouse.
3. T1 was stopped immediately after reading the new_ORDER table for the selected warehouse and district. No qualifying row was found.
4. A New-Order transaction T2 was started for the same warehouse and district. T2 completed and was committed without being blocked by T1.
5. T1 was resumed and the new_ORDER table was read again. No qualifying row was found.
6. T1 completed and was committed.
7. The NO_D_ID of all new_ORDER rows for the selected warehouse and district was restored to the original value. The changes were committed.

4.5 Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transaction 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)…*

Specified durability tests were executed to demonstrate satisfaction of the durability requirements for this implementation of TPC Benchmark C. One durability test, described below, covering the following failure situations was performed:

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (Clause 3.5.3.1).*

This test was performed under a load of 37,440 users on the full-scale database built for 74,880 users. Another durability test, described below, combining the following failure situations was performed:

- *instantaneous interruption which requires system reboot [of processors] to recover. (Clause 3.5.3.2*)
- *failure of all or part of memory. (Clause 3.5.3.3).*

This test was performed under the full performance-measurement load of 74,880 users.

4.5.1 Loss of Data Disk or Log Disk

Because the log and data-storage devices are Redundant Disk Arrays which each function independently of the rest of the system in ensuring data integrity under loss and/or replacement of any individual disk drive (and other failures as well), integrity under such failure and replacement does not entail any interruption in processing. The test below validates the durability by demonstrating persistence of the results of transactions processed both before and during these failures, validating the durability upon database recovery (in this instance, forced) of transactions which completed before the failure and the non-effect of transactions which did not complete.

1. The D_NEXT_O_ID fields for all rows in the DISTRICT table were summed up to determine the initial count of the total number of orders (count1).
2. A test was initiated with 37,440 terminals. On the driver system, completed/rolled-back transactions (including New-Orders) were recorded in a "success" file.
3. After 10 minutes, one of the individual disks containing Oracle system tablespace and one containing recovery log were each unplugged from its array. On the system console messages appeared indicating that the data from the missing disk was being rebuilt on other disks, using the redundancy features of the array. However, system processing continued normally.
4. The test finished normally.
5. Step 1 was repeated to determine the total number of orders (count2). Count2-count1 was the same as the number of records for successful New Orders in the RTE "success" file.
6. Consistency checks 1-4 were run before and after the benchmark run and the results were verified.

4.5.2 Instantaneous Interruption and Loss of Memory

Instantaneous interruption and loss of memory tests were combined because the loss of power erases the contents of memory. This failure was induced while the benchmark was running by turning off the power supplies to the server.

1. The D_NEXT_O_ID fields for all rows in district table were summed up to determine the initial count of the total number of orders (count1).
2. Transactions were started at full load. On the driver system, completed/rolled-back transactions (including New-Orders) were recorded in a "success" file.
3. After ten minutes the server systems were de-powered.
4. The test was aborted on the driver.
5. The server system was restarted.
6. The database was restarted and a recovery performed using the transaction log.
7. The contents of the "success" file on the driver and the ORDERs table were spot-compared to verify that records in the "success" file for completed New-Order transactions had corresponding records in the ORDERs table.
8. Step 1 was repeated to determine the current total number of orders (count2). Count2-count1 (= 626,109) was 12 more than the number of records for successful New Orders in the RTE "success" file (= 632,596 – 6,499 rolled-back =626,097). *This difference would be due only to transactions which were committed on the system under test but for which the output data was not displayed on the [emulated] input/output screen before the failure*.
9. Consistency checks 1-4 were run before and after the benchmark run and the results were verified.

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, must be disclosed. If the database was overscaled and inactive rows of the WAREHOUSE table were deleted the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The TPC-C database for this test was configured with 7,488 warehouses.

Table	Occurrences
Warehouse	7,488
District	74,880
Customer	224,640,000
History	224,640,000
Orders	224,640,000
New Orders	67,392,000
Order Line	2,246,049,936
Item	100,000
Stock	7,488,000,000

5.2 Database and Growth Layout

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

Table 5.2 indicates the distribution of the database tables over the disks of the tested and priced systems**.**

Table 5.2: Disk Usage in Tested System

Table Chunk	Disk Address	LVM Usage	Allocated Space (MB)
OS+SWAP	c1t6d0		2048
/project	c15t12d0		2048
/BACKUP	c15t12d1		2048
/MISC	c15t12d2		2048
/UTIL	c15t12d3		2048
Unused	c15t12d4		2048
log1	c15t12d5		10000
log2	c13t15d2		10000
log3	c13t15d3		10000
log4	c13t14d0		10000
log5	c13t14d1		10000
log.bk	vgtpcc	15 way striped	10000
SWAP	vgtpcc	15 way striped	16834
master	vgtpcc	15 way striped	320
master.bk	vgtpcc	15 way striped	320
tmpdb	vgtpcc	15 way striped	128
tmpdb.bk	vgtpcc	15 way striped	128
w_d_i_no	vgtpcc	15 way striped	600
w_d_i_no.bk	vgtpcc	15 way striped	600
history1	vgtpcc	15 way striped	9000
history1.bk	vgtpcc	15 way striped	9000
order_line1	vgtpcc	40 way striped	8600
order_line1.bk	vgtpcc	40 way striped	8600
order_line2	vgtpcc	40 way striped	8600
order_line2.bk	vgtpcc	40 way striped	8600
order_line3	vgtpcc	40 way striped	8600
order_line3.bk	vgtpcc	40 way striped	8600

order_line4	vgtpcc	40 way striped	8600
order_line4.bk	vgtpcc	40 way striped	8600
order_line5	vgtpcc	40 way striped	8600
order_line5.bk	vgtpcc	40 way striped	8600
order_line6	vgtpcc	40 way striped	8600
order_line6.bk	vgtpcc	40 way striped	8600
order_line7	vgtpcc	40 way striped	8600
order_line7.bk	vgtpcc	40 way striped	8600
order_line8	vgtpcc	40 way striped	8600
order_line8.bk	vgtpcc	40 way striped	8600
order_line9	vgtpcc	40 way striped	8600
order_line9.bk	vgtpcc	40 way striped	8600
order_line10	vgtpcc	40 way striped	8600
order_line10.bk	vgtpcc	40 way striped	8600
order_line11	vgtpcc	40 way striped	8600
order_line11.bk	vgtpcc	40 way striped	8600
order_line12	vgtpcc	40 way striped	8600
order_line12.bk	vgtpcc	40 way striped	8600

Table Chunk	Disk Address	LVM Usage	Allocated Space (MB)
orders1	vgtpcc	40 way striped	375
orders1.bk	vgtpcc	40 way striped	375
orders2	vgtpcc	40 way striped	375
orders2.bk	vgtpcc	40 way	375

		striped	
orders3	vgtpcc	40 way	375
		striped	
orders3.bk	vgtpcc	40 way	375
		striped	
orders4	vgtpcc	40 way	375
		striped	
orders4.bk	vgtpcc	40 way	375
		striped	
orders5	vgtpcc	40 way	375
		striped	
orders5.bk	vgtpcc	40 way	375
		striped	
orders6	vgtpcc	40 way	375
		striped	
orders6.bk	vgtpcc	40 way	375
		striped	
orders7	vgtpcc	40 way	375
		striped	
orders8	vgtpcc	40 way	375
		striped	
orders8.bk	vgtpcc	40 way	375
		striped	
orders9	vgtpcc	40 way	375
		striped	
orders9.bk	vgtpcc	40 way	375
		striped	
orders10	vgtpcc	40 way	375
		striped	
orders10.bk	vgtpcc	40 way	375
		striped	
orders11	vgtpcc	40 way	375
		striped	
orders11.bk	vgtpcc	40 way	375
		striped	
orders12	vgtpcc	40 way	375
		striped	
orders12.bk	vgtpcc	40 way	375
		striped	
C_index1	vgtpcc	40 way	1200
		striped	
C_index1.bk	vgtpcc	40 way	1200
		striped	
C_index2	vgtpcc	40 way	1200
		striped	
C_index2bk	vgtpcc	40 way	1200

		striped	
C_index3	vgtpcc	40 way striped	1200
C_index3.bk	vgtpcc	40 way striped	1200
C_index4	vgtpcc	40 way striped	1200
C_index4.bk	vgtpcc	40 way striped	1200
C_index5	vgtpcc	40 way striped	1200
C_index5.bk	vgtpcc	40 way striped	1200
C_index6	vgtpcc	40 way striped	1200
C_index6.bk	vgtpcc	40 way striped	1200
C_index7	vgtpcc	40 way striped	1200
C_inde7.bk	vgtpcc	40 way striped	1200
C_index8	vgtpcc	40 way striped	1200
C_index8.bk	vgtpcc	40 way striped	1200
C_index9	vgtpcc	40 way striped	1200
C_index9.bk	vgtpcc	40 way striped	1200
C_index10	vgtpcc	40 way striped	1200
C_index10.bk	vgtpcc	40 way striped	1200

Table Chunk	Disk Address	LVM Usage	Allocated Space (MB)
customer1	c10t1d1		2540
customer2	c10t4d1		2540
customer3	c10t5d1		2540
customer4	c10t6d1		2540
customer5	c12t2d1		2540
customer6	c12t3d1		2540
customer7	c12t5d1		2540
customer8	c12t6d1		2540
customer9	c16t1d1		2540
customer10	c16t2d1		2540

customer11	c16t3d1		2540
customer12	c16t4d1		2540
customer13	c17t2d1		2540
customer14	c17t3d1		2540
customer15	c17t5d1		2540
customer16	c17t6d1		2540
customer17	c19t1d1		2540
customer18	c19t4d1		2540
customer19	c19t5d1		2540
customer20	c19t6d1		2540
customer21	c20t1d1		2540
customer22	c20t2d1		2540
customer23	c20t3d1		2540
customer24	c20t4d1		2540
customer25	c21t1d1		2540
customer26	c21t4d1		2540
customer27	c21t5d1		2540
customer28	c21t6d1		2540
customer29	c9t1d1		2540
customer30	c9t2d1		2540
customer31	c9t3d1		2540
customer32	c9t4d1		2540
customer33	c5t1d1		2540
customer34	c5t4d1		2540
customer35	c5t5d1		2540
customer36	c5t6d1		2540
customer37	c7t2d1		2540
customer38	c7t3d1		2540
customer39	c7t5d1		2540
customer40	c7t6d1		2540

Table Chunk	Disk Address	LVM Usage	Allocated Space (MB)
stock1	c13t11d0		3951
stock2	c7t5d0		3951
stock3	c7t3d0		3951
stock4	c7t2d0		3951
stock5	c5t6d0		3951
stock6	c5t5d0		3951
stock7	c5t4d0		3951
stock8	c5t1d0		3951
stock9	c9t4d0		3951
stock10	c9t3d0		3951
stock11	c9t2d0		3951
stock12	c9t1d0		3951
stock13	c21t6d0		3951

stock14	c21t5d0	3951
stock15	c21t4d0	3951
stock16	c21t1d0	3951
stock17	c20t4d0	3951
stock18	c20t3d0	3951
stock19	c20t2d0	3951
stock20	c20t1d0	3951
stock21	c19t6d0	3951
stock22	c19t5d0	3951
stock23	c19t4d0	3951
stock24	c19t1d0	3951
stock25	c17t6d0	3951
stock26	c17t5d0	3951
stock27	c17t3d0	3951
stock28	c17t2d0	3951
stock29	c15t13d2	3951
stock30	c16t3d0	3951
stock31	c16t2d0	3951
stock32	c16t1d0	3951
stock33	c12t6d0	3951
stock34	c12t5d0	3951
stock35	c12t3d0	3951
stock36	c12t2d0	3951
stock37	c10t6d0	3951
stock38	c10t5d0	3951
stock39	c10t4d0	3951
stock40	c10t1d0	3951

The distribution of the database tables over the 106 disk arrays of the priced system is an extension of the distribution described in Table 5.2; some ancillary details are mentioned in Appendix E. 180-day storage growth requirements are met with the unused space of this configuration. Figure 1.2 shows the configuration of the priced-system disks.

5.3 Data Model & Interfaces

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 used (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.*

Oracle8i Enterprise Database Server v8.1.5 is a relational DBMS. SQL stored procedures were used, invoked through the Oracle Call Interface (OCI); the application code appears in Appendix A.

5.4 Partitions/Replications

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

No partitioning or replication was used.

5.5 Growth Requirements

Details of the 180 day space computations along with proof that the database is configured to sustain 8 hours for the dynamic tables (Order, Order-Line, and History) must be disclosed.

See Appendix E.

6 Clause 5 Related Items

6.1 Throughput

Measured tpmC must be reported.

Table 6.1: Measured tpmC

tpmC®	92,832.96

6.2 Response Time

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

Table 6.2: Response Times

Response Times	Average	90th %-ile	Maximum
New-Order	0.67s	1.14s	4.55s
Payment	0.56s	1.02s	4.41s
Order-Status	0.62s	1.09s	3.54s
Delivery (interactive portion)	0.07s	0.08s	0.12s
Delivery (deferred portion)	0.72s	1.19s	4.33s
Stock-Level	0.79s	1.38s	4.43s
Menu	0.001s	0.10s	0.53s

6.3 Keying and Think Times

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

Table 6.3: Keying Times

Keying Times	Minimum	Average	Maximum
New Order	18.02s	18.03s	18.04s
Payment	3.01s	3.02s	3.03s
Order Status	2.01s	2.02s	2.03s
Interactive Delivery	2.01s	2.02s	2.03s
Stock Level	2.01s	2.02s	2.02s

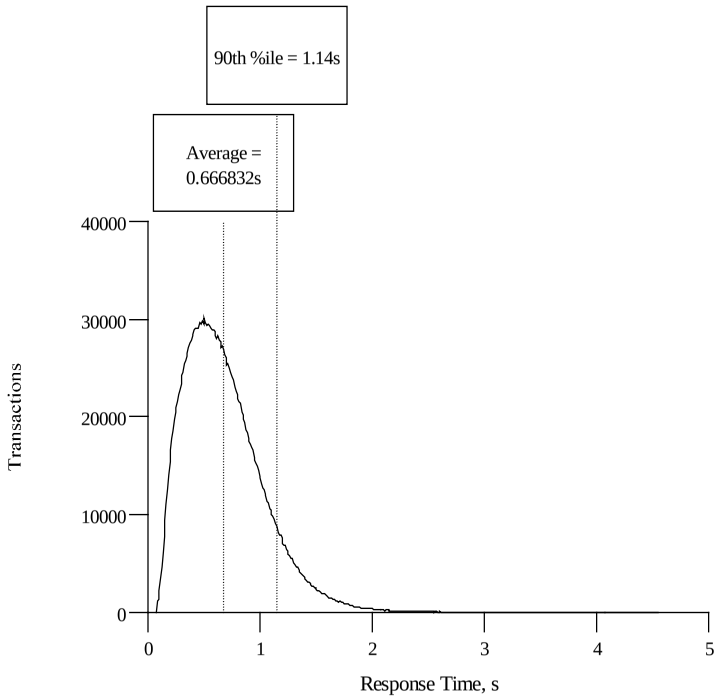
Table 6.4: Think Times

Think Times	Minimum	Average	Maximum
New Order	0.01s	12.13s	191.14s
Payment	0.01s	12.06s	173.28s
Order Status	0.01s	10.06s	119.67s
Interactive Delivery	0.02s	5.07s	61.74s
Stock Level	0.01s	5.07s	53.89s

6.4 Response Time Frequency Distribution Curves and Other Graphs

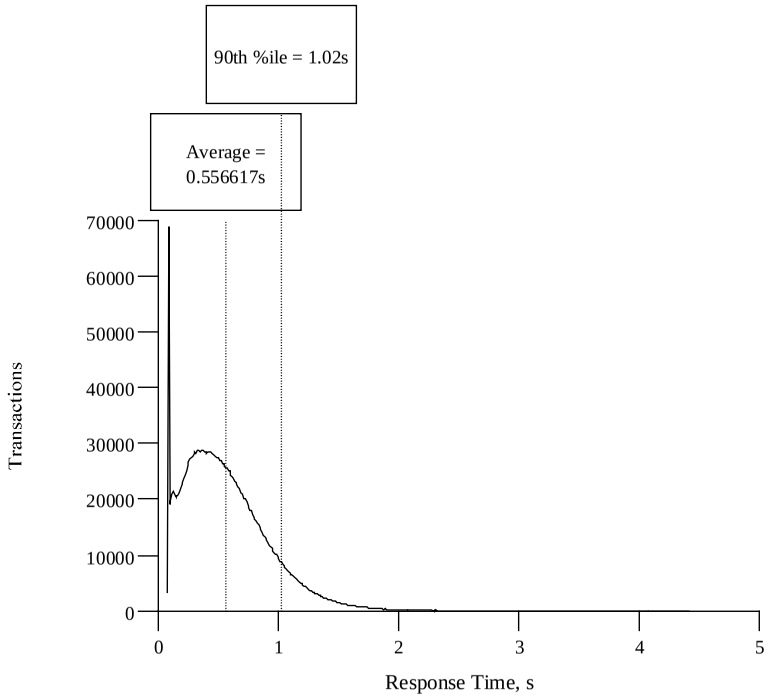
Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. The Think Time frequency distribution curve (see Clause 5.6.3) must be reported for the New-Order transaction. A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction, and the measurement interval indicated.

Figure 6.1: New Order Response Time Distribution



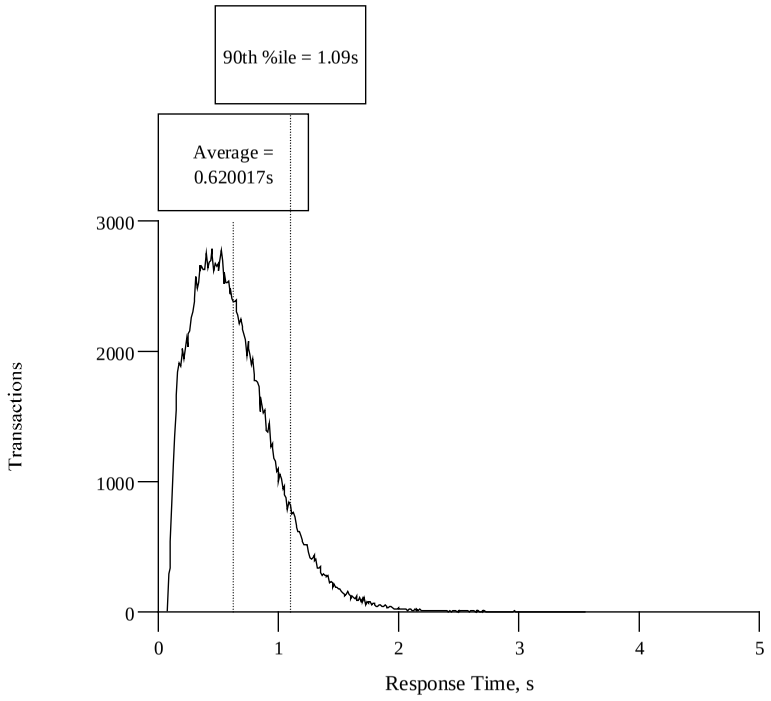
Response time frequency distribution for New Order transaction

Figure 6.2: Payment Response Time Distribution



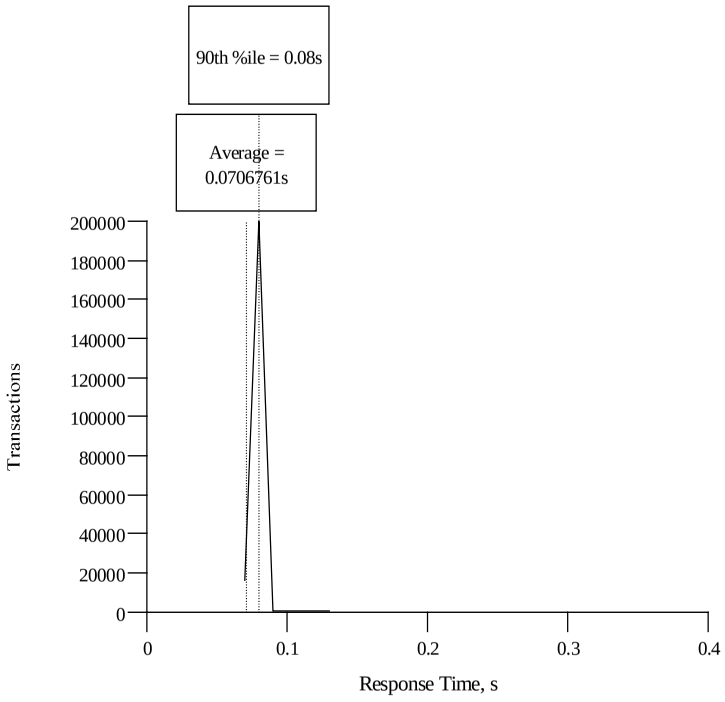
Response time frequency distribution for Payment transaction

Figure 6.3: Order Status Response Time Distribution



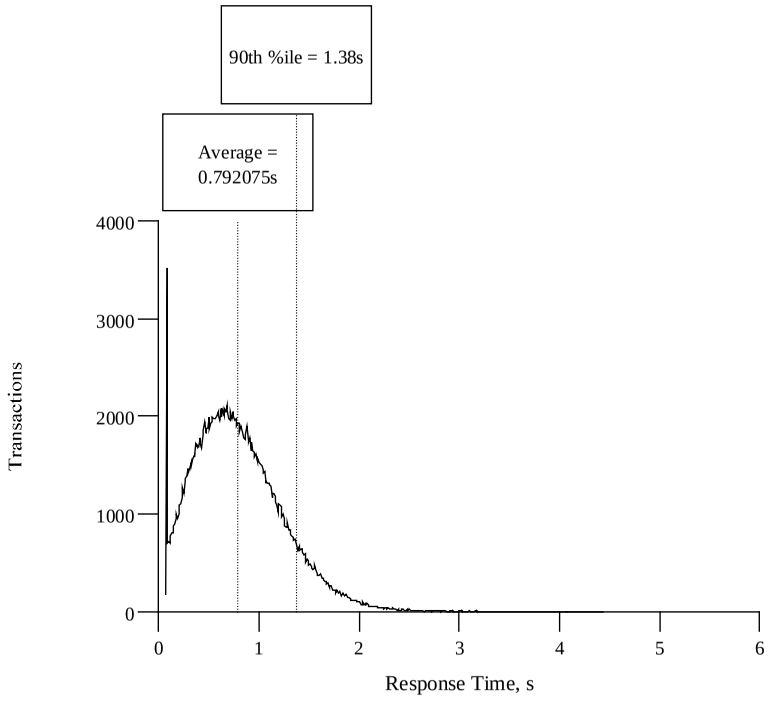
Response time frequency distribution for Order Status transaction

Figure 6.4: (Interactive) Delivery Response Time Distribution



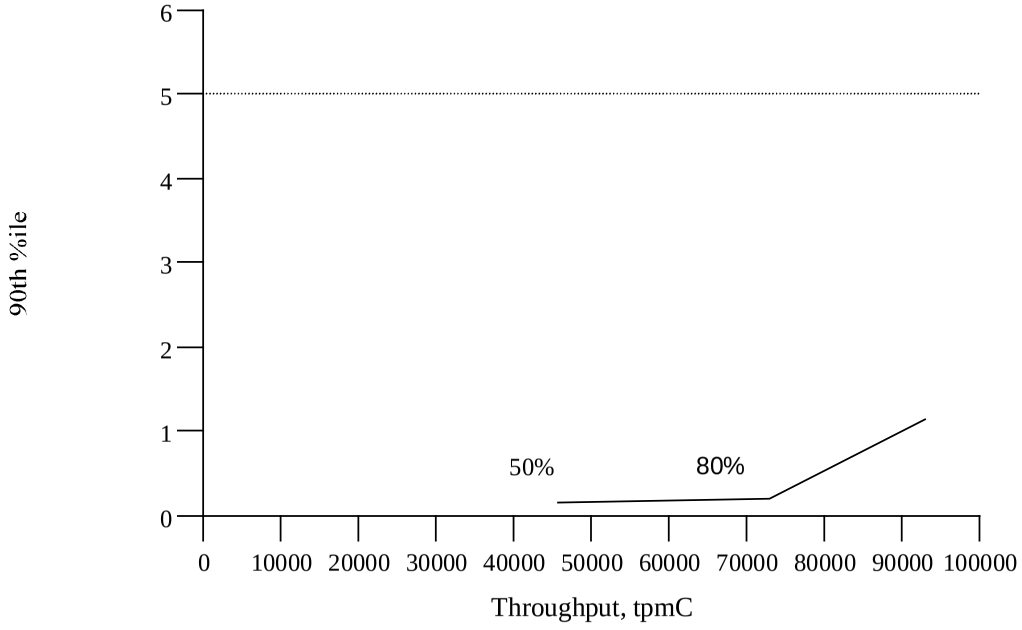
Response time frequency distribution for Delivery transaction

Figure 6.5: Stock Level Response Time Distribution



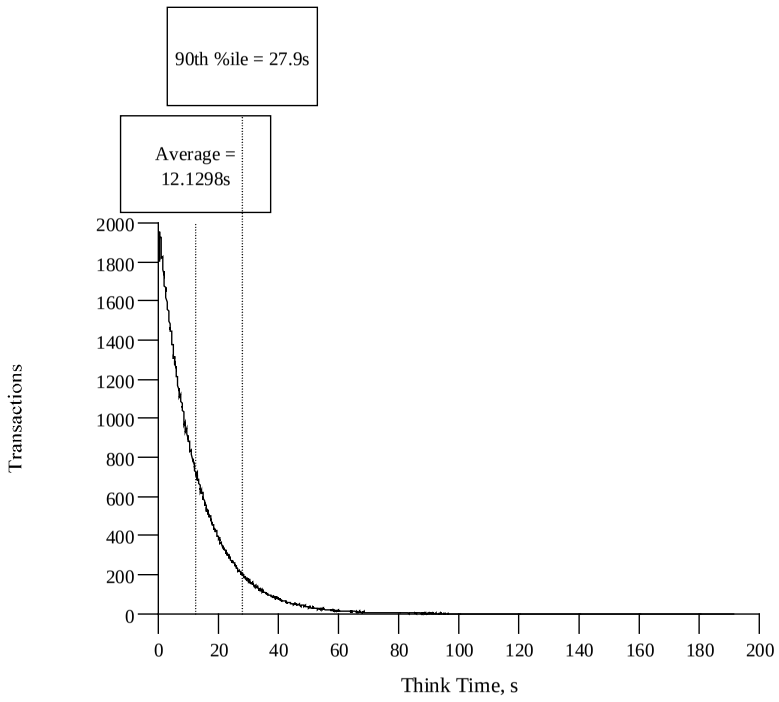
Response time frequency distribution for Stock Level transaction

Figure 6.6: Response Time Versus Throughput



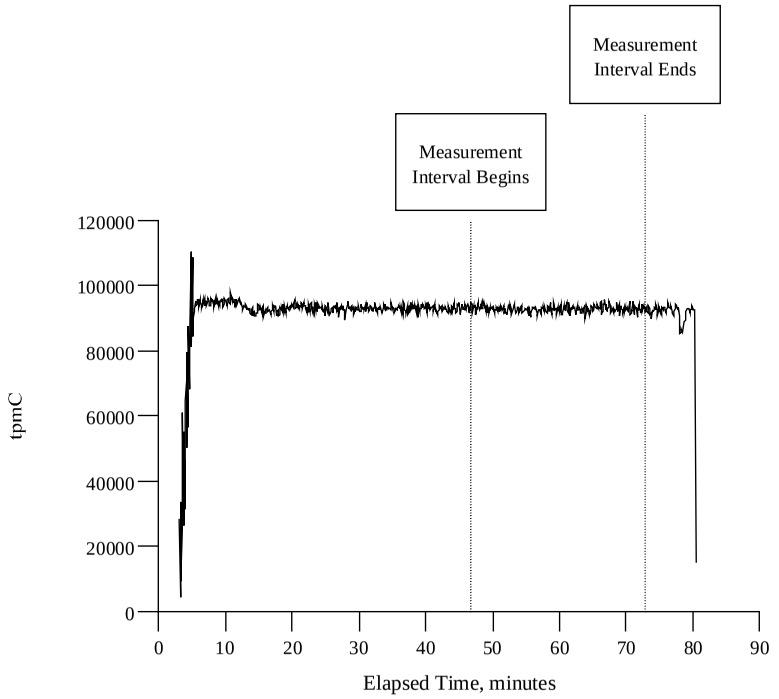
New Order response time versus Throughput

Figure 6.7: New Order Think Time Distribution



Think time frequency distribution for New Order transaction

Figure 6.8: Throughput Versus Time



Throughput of the New-Order transaction versus elapsed time

6.5 Steady State Determination

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

Synchronization techniques employed in the benchmark process ensure that all emulated users are logged into the client and have opened the application before submitting transactions. Once all users are connected, each pauses a random amount of time before submitting transactions. The pause time distribution is controlled by a benchmark input parameter. The ramp-up interval is discernible in the graph of throughput over time. The data reduction also tracks the user load and indicates the point in time at which all users have submitted at least one transaction. The throughput is observed to be steady within the systematic and statistical variability of the measurement after all users are submitting transactions. A checkpoint is initiated upon the end of ramp-up.

6.6 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.

Modified database buffers migrated to disk on a least-recently-used basis independent of transaction commits. In addition, every block modification was protected by redo log records. These redo log records were written to the redo log buffer (in memory) and were flushed to a redo log file on disk either when the transaction committed or when the redo log buffer became full. However, due to the rapid commit during this benchmark, the redo log buffer was always flushed by a commit before it became full. Also, because many transactions were committing in a short period of time, a single flush of the redo log buffer resulted in many transactions' redo log data being written to disk. This is called group commit.

6.6.1 Checkpoint

During an Oracle8i Enterprise Database Server v8.1.5 checkpoint, all modified blocks in the shared buffer cache which had not been written to disk since the last checkpoint are written to disk.

6.6.2 Checkpoint Conditions

Oracle8i Enterprise Database Server v8.1.5 performs a checkpoint for the following conditions:

1. A redo log switch occurs.
2. The amount of data written to a redo log reaches the log_checkpoint_interval
3. The amount of time since the last checkpoint reaches the log_checkpoint_timeout.

6.6.3 Checkpoint Implementation

Checkpoints were allowed to occur whenever a redo log filled up and caused a redo log switch. The checkpoint interval during the steady state phase was 26 minutes.

6.6.4 Serializable Transactions

Oracle supports serializable transaction isolation in full compliance with the SQL92 and TPC-C requirements. This is implemented by extending the multiversity concurrency control mechanism long supported by Oracle.

Oracle queries take no read locks and see only data committed as of the beginning of the query's execution. This means that readers and writers coexist without blocking one another, providing a high degree of concurrency and consistency. While this mode does prevent reading dirty data, Oracle's default isolation level also permits a transaction that issues a query twice to see non-repeatable reads and phantoms, as defined in SQL92 and TPC-C.

Beginning with Oracle7 release 7.3, a transaction may request a high degree of isolation with the command SET TRANSACTION ISOLATION LEVEL SERIALIZABLE, as defined in SQL92. This transaction mode prevents read/write and write/write conflicts that would cause serializability failures. A session can establish this mode as its default mode, so the SET TRANSACTION command need not be issued in each transaction.

Oracle implements SERIALIZABLE mode by extending of the scope of read consistency from the individual query to the entire transaction. Instead of limiting a query to data committed at the time a query begins, in a serializable transaction all queries see data as of the beginning of the transaction. Thus, a serializable transaction sees a fixed snapshot of the database, established as of the beginning of the transaction.

All reads within a serializable transaction see only committed data as of the start of that transaction, plus new updates done by the transaction itself. All reads by a serializable transaction are therefore repeatable, as the transaction will access prior versions of data changed (or deleted) by other transactions after the start of the serializable transaction. This behavior also results in phantom protection since new rows created by other transactions will be invisible to the serializable transaction.

To ensure proper isolation, a serializable transaction cannot modify rows that were changed by other transactions after the beginning of the serializable transaction. If a serializable transaction attempts to update (or delete) a row previously changed by another transaction (serializable or not) since the beginning of the serializable transaction, the update (or delete) statement will fail with error ORA-08177: "Can't serialize access", and the statement will rollback.

SET TRANSACTION ISOLATION

LEVEL SERIALIZABLE;

SELECT …

SELECT…

UPDATE…

IF "Can't serialize access"

THEN ROLLBACK; LOOP and retry

ELSE COMMIT;

When a serializable transaction fails with this error, the application may either commit the work executed to that point, execute additional (but different) statements, or rollback the entire transaction. Repeated attempts to execute the same statement will always fail with the error "Can't serialize access", unless the other transaction has rolled back and released its lock. This error and these recovery options are similar to the treatment of deadlocks in systems that use read locks to ensure serializable execution. In both cases, conflicts between transactions cannot be resolved unless one of the transactions rolls back and restarts or commits without re-executing the statement receiving the error.

6.7 Reproducibility

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

A second measurement achieved a qualified throughput of 92777.38 tpmC over a 26-minute, steady-state interval.

6.8 Measurement Period Duration

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

The measurement interval was 26 minutes.

6.9 Regulation of Transaction Mix

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 selection method of *Clause 5.2.4.1* was used. The weights were not adjusted during the run.

6.10 Transaction Mix

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

Table 6.5: Transaction Mix

Type	Percentage
New Order	44.79%
Payment	43.06%
Order Status	4.05%
Delivery	4.05%
Stock Level	4.05%

6.11 Transaction Statistics

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order trasaction must be disclosed. The percentage of remote order-lines entered per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

See Table 3.1

6.12 Checkpoint Count and Location

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.

Times in Table 6.6 below are relative to the beginning of the driver-timed phase of the test. The checkpoint within the 26-minute measurement interval started 552 seconds from its start. The checkpoint interval during steady-state is 26 minutes. In accord with 5.5.2.2, there is no checkpoint within the guard zones 1560/4=390 seconds from the beginning and end of the measurement interval.

Table 6.6: Checkpoints

Event	From (s)	To (s)	Duration (s)
Checkpoint	1783	2380	597
Measured Interval	2805	4365	1560
Checkpoint	3357	3955	598
	5346	5649	303

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 input parameters, code fragments, functions, et cetera used to generate each transaction input field must be disclosed. Comment: The intent is to demonstrate the RTE was configured to generate transaction input data as specified in Clause 2.

The RTE (Remote Terminal Emulator) on the driver system was developed at Hewlett-Packard and is not commercially available. Appendix D lists RTE input parameters and code fragments used to generate each transaction input field.

For this instance of the TPC-C benchmark, 12 drivers and 24 clients were used. The drivers emulated users logged in to the clients. An overview of the benchmark software on the drivers, clients and server is shown in Figure 7.1.

The benchmark is started with the **run** command on the driver system. **Run** controls the overall execution of the benchmark. After reading a configuration file, **run** starts TUXEDO on the client, collects pre-benchmark audit information and inserts a timestamp into a database audit table. When all the initial steps are completed, **run** invokes another program, **driver,** to start the benchmark. As the benchmark completes, **run** shuts down TUXEDO and collects the benchmark results into a single location.

Driver is the heart of the benchmark software. It simulates users as they log in, execute transactions and view results. **Driver** collects response times for each transaction and saves them in a file for future analysis.

Qualify is the post-processing analysis program. It produces the numerical summaries and histograms needed for the disclosure report.

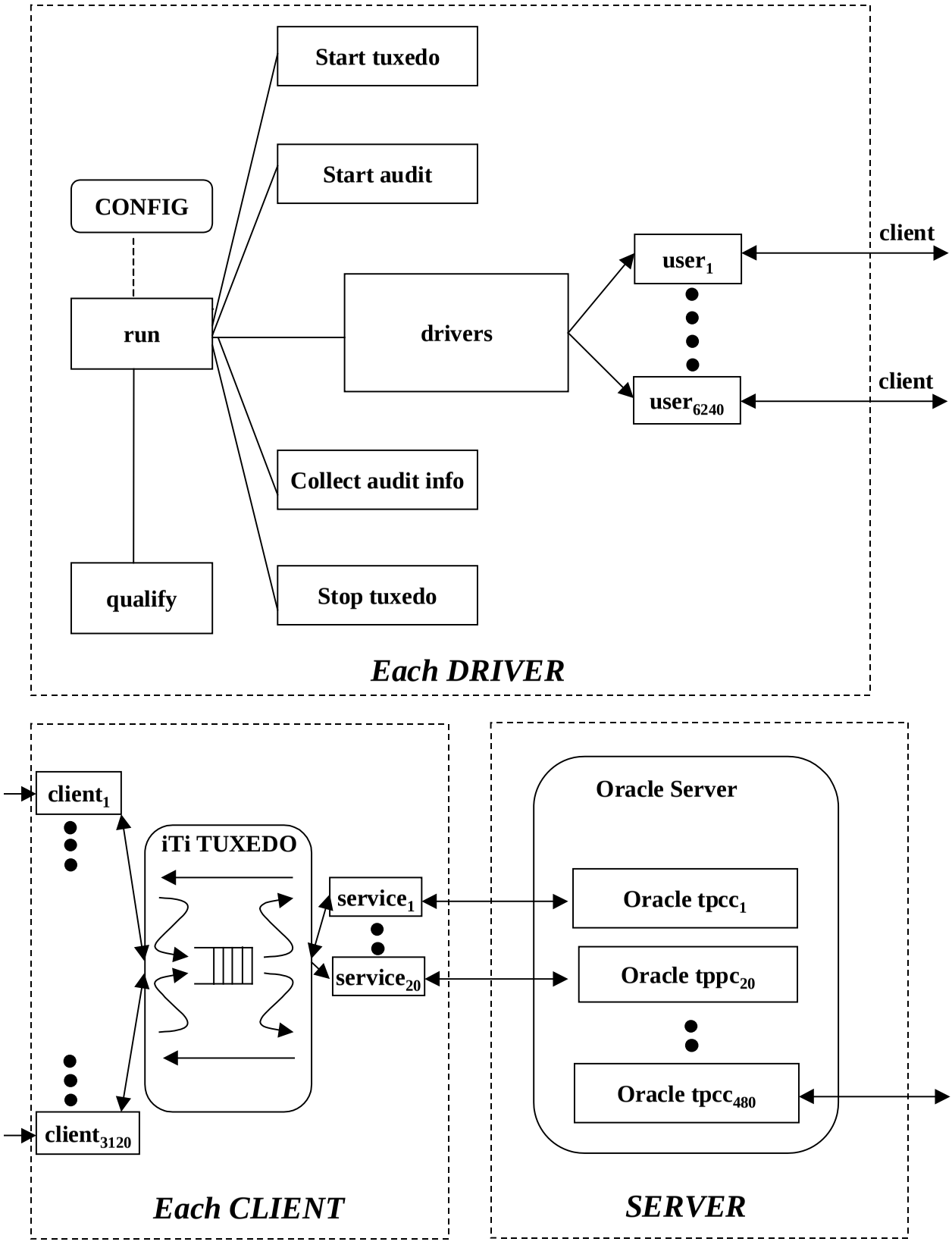


Figure 7.1: Benchmark Software

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 the priced system.

In the priced configuration, workstations are connected to the clients via LANs. On the tested system, 12 LAN segments carried all the traffic between the 74880 simulated users in the RTE system and the 24 client systems. In the priced configuration, this traffic has been divided among 4 separate LAN segments for each of the 24 clients, for a total of 96 LAN segments.

After the initial publication of this FDR, it was brought to our attention that we had used 100BaseT links between the driver systems and the clients but had priced 10BaseT hubs. This was done purely as a matter of simplifying the very large tested configuration, not to gain a performance advantage.

In the priced configuration, the 3,120 users assigned to each client were attached via 4 10BaseT lan segments (the priced --and tested-- network cards on the clients are self-sensing 10BaseT/100BaseT cards). However, in the measured configuration, the load for two clients (6,240 users) was generated by one driver connected to the switch via a single 100BaseT lan. This simplification was made to reduce the physical amount of cabling and the number of network cards on the driver systems, it was not intended to enhance performance.

To prove that this substitution was performance-neutral, we have conducted the following test to satisfy Clauses 6.6.3.4 (only so far as showing the performance of the emulated system is the same as the priced system; the driver systems do not perform any functions beyond that of an RTE), 6.6.4, 8.1.1.7, 8.1.7.2, and 8.1.7.4.

In this test, the 100BaseT links on 2 driver systems were replaced by six 10BaseT links on each of these 2 drivers . Note that in the priced configuration, the load of 6,240 users flows over 8 10BaseT links; so, the number of priced 10BaseT lan segments (8 per 6,240 users) was more than what was configured on these 2 driver systems (6 per 6,240 emulated users). The other 10 driver systems remained connected via a single 100BaseT link each. The following table shows the germane performance indices for the 12 driver systems, as well as the aggregates for the whole system.

Note that this was a qualified run with a qualified measurement period, etc.

Performance index		RTE 1	RTE 2	RTE 3	RTE 4	RTE 5	RTE 6	RTE 7	RTE 8	RTE 9	RTE 10	RTE 11	RTE 12	Aggregate
Qualified throughput		7695.62	7732.19	7700.42	7750.46	7747.96	7753.58	7751.12	7777.27	7741	7763.96	7766.73	7765.38	92945.69
New Order response time	90th %	1.23	1.16	1.21	1.13	1.09	1.11	1.11	0.95	1.08	1.01	0.94	1.02	1.1
	Avg	0.74	0.7	0.73	0.63	0.62	0.63	0.66	0.56	0.63	0.58	0.55	0.58	0.63
	Max	3.65	3.27	3.3	3.28	3.34	3.45	2.85	3.25	3.32	7.36	3.35	3.35	7.36
Payment response time	90th %	1.12	1.04	1.1	1.02	0.97	1	0.98	0.83	0.96	0.89	0.83	0.9	0.98
	Avg	0.63	0.59	0.62	0.52	0.51	0.52	0.54	0.44	0.52	0.47	0.44	0.47	0.52
	Max	3.29	3.13	3.65	3.28	3.23	3.26	2.82	2.72	3.39	7.27	3.11	3.22	7.27
Order Status response time	90th %	1.18	1.11	1.16	1.09	1.04	1.07	1.06	0.91	1.04	0.96	0.89	0.97	1.05
	Avg	0.69	0.65	0.69	0.59	0.58	0.59	0.61	0.51	0.59	0.54	0.51	0.54	0.59
	Max	3.19	2.81	2.79	2.67	2.99	3.05	2.56	2.57	3.28	6.54	2.67	2.52	6.54
Delivery response time	90th %	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
	Avg	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
	Max	0.12	0.12	0.12	0.12	0.12	0.12	0.13	0.13	0.12	0.12	0.12	0.12	0.13
Stock Level response time	90th %	1.49	1.41	1.47	1.38	1.32	1.36	1.36	1.22	1.33	1.27	1.2	1.27	1.35
	Avg	0.87	0.83	0.86	0.76	0.73	0.76	0.78	0.67	0.75	0.71	0.67	0.71	0.76
	Max	3.57	3	3.87	3.32	3.24	3.34	2.92	3.09	3.41	3.05	3.2	2.92	3.87
New Order Menu response time	Min	0	0	0	0	0	0	0	0	0	0	0	0	0
	Avg	0	0	0	0	0	0	0	0	0	0	0	0	0
	Max	0.06	0.06	0.06	0.06	0.06	0.05	0.06	0.52	0.05	0.05	0.05	0.05	0.52
Payment Menu response time	Min	0	0	0	0	0	0	0	0	0	0	0	0	0
	Avg	0	0	0	0	0	0	0	0	0	0	0	0	0
	Max	0.06	0.06	0.06	0.06	0.06	0.05	0.06	0.06	0.05	0.05	0.05	0.05	0.06
Order Status Menu response time	Min	0	0	0	0	0	0	0	0	0	0	0	0	0
	Avg	0	0	0	0	0	0	0	0	0	0	0	0	0
	Max	0.06	0.06	0.06	0.05	0.06	0.05	0.06	0.06	0.05	0.05	0.05	0.05	0.06
Delivery Menu response time	Min	0	0	0	0	0	0	0	0	0	0	0	0	0
	Avg	0	0	0	0	0	0	0	0	0	0	0	0	0
	Max	0.06	0.06	0.06	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06
Stock Level Menu response time	Min	0	0	0	0	0	0	0	0	0	0	0	0	0
	Avg	0	0	0	0	0	0	0	0	0	0	0	0	0
	Max	0.06	0.06	0.05	0.05	0.06	0.05	0.06	0.06	0.05	0.05	0.05	0.05	0.06

It is difficult to detect that the two driver systems that used the 10BaseT links were numbers 7 and 8. This proves that the substitution of a 100BaseT link for eight 10BaseT links did not produce a performance gain and the tested system was a correct emulation of the priced system.

7.3 Functional Diagrams

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

Figures 1.1 and 1.2 (in Chapter 1) show functional diagrams of the benchmark and configured systems. A description of the RTE and benchmark software is provided above.

7.4 Networks

The network configuration of both the tested and proposed services which are being represented and a thorough explanation of exactly which parts are being replaced with the Driver System must be disclosed.

Figures 1.1 and 1.2 (in Chapter 1) diagram the network configurations of the benchmark and configured systems, and represent the Driver connected via LAN replacing the workstations and HUBs connected via LANs. The clients are connected via 100 Base-T to a ProCurve Switch 4000M which in turn is connected via 1000 Base-T Ethernet to the SUT.

The bandwidth of the networks used in the tested/priced configurations must be disclosed.

Ethernet and 100 Base-T local area networks (LAN) with a bandwidth of 100 megabits per second are used in the tested/priced configurations. The 1000BT used has a bandwidth of 1000 megabits per second.

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 item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery data. If package-pricing is used contents of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.

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.

Each priced configuration consists of an integrated system package, additional options, and components. Prices for all Hewlett-Packard products are US list prices. A one (1) year warranty is standard with all Hewlett-Packard products.

8.2 Support Pricing

The five year support pricing for Hewlett-Packard products is based on forty-eight (48) months of monthly support costs; sixty (60) months minus the twelve month warranty period. The Oracle Corporation support pricing is based on sixty (60) months of monthly support costs. The following support products were priced in the benchmark:

- HP four-hour on-site repair hardware support,
- HP telephone support for software and updates
- Oracle Corporation Standard Technical Support and,
- BEA TUXEDO Standard Technical Support

8.2.1 HP Hardware Support

HP four-hour maximum response, on-site support for hardware provides service from 8:00 A.M. to 5:00 P.M. Monday through Friday. Service requests made as late as 5:00 P.M. will receive a response the same day.

8.2.2 HP Software Support

HP Software Support provides the following:

- Access to the HP Response Centers for fault isolation and problem solving assistance,
- Guaranteed two (2) hour call return, immediate response for critical calls,
- Electronic access to product and support information,
- Electronic access to software patches,
- Right-to-use and copy software updates.

8.2.3 Hubs

An additional 10% of the needed hubs were included in the priced configuration to provide the required four hour repair for hardware components. The return-for-replacement support would be used to restock spares.

8.3 Oracle Corporation Standard Technical Support

Oracle Corporation Standard Technical Support includes:

Product updates,

- A regular technical publication,
- Unlimited, toll-free telephone service to assist in product installation, syntax, and usage that is available from 7:00 A.M. to 5:30 P.M. PST Monday through Friday.

8.4 Availability

The committed delivery date for general availability (availability date) of products used in the price calculation 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.

see below

8.5 Priced System Configuration

The hardware, software, and support/maintenance products priced in this benchmark are detailed on page *v*.

8.6 Throughput, Price/Performance, and Availability Date

A statement of the measured tpmC® *as well as the respective calculations for the 5-year pricing, price/performance (price/tpmC®).*

For Throughput and Price/Performance, please see page iv and v. The Price/Performance calculation spreadsheet appears on page v.

All hardware components in this test of the HP 9000 V2500 Enterprise Server system are currently available. HP-UX 11.0 64-bit incorporating Extension Pack 9903 will be available on June 30, 1999. Oracle8i Enterprise Database Server v8.1.5 will be available on August 31, 1999.

9 Clause 9 Related Items

9.1 Auditor's Report

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

If audited, the auditor's attestation letter must be made readily available to the public as part of the Full Disclosure Report, but a detailed report from the auditor is not required.

This implementation of the TPC Benchmark® C on the HP 9000 V2500 Enterprise Server was audited by Tom Sawyer for Performance Metrics, Inc..

Tom Sawyer
Performance Metrics, Inc.
2229 Benita Drive, Suite 101
Rancho Cordova, CA 95670
U.S.A.
Phone: 916 635-2822
Fax: 916 858-0109

The attestation letter is shown on the following pages.

March 1, 1999

Mr. Mark Kroot
High Performance Systems Division
Hewlett-Packard Company
19111 Pruneridge Avenue
Cupertino, CA 95014

I have verified the TPC Benchmark™ C client/server for the following configuration:

Platform: **Hewlett-Packard 9000 V2500 Enterprise Server**
Database Manager: **Oracle8 Enterprise Database Server v8*i***
Operating System: **HP-UX 11.00 64-bit Extension Pack 990P**
Transaction Manager: **Tuxedo version 6.4**

Server: Hewlett-Packard 9000 V2500 Enterprise Server				
CPU's	**Memory**	**Disks**	**90% Response**	**tpmC**
32 PA-8500 @ 440 MHz	**Main: 32 GB iCache: 512KB each dCache: 1MB each**	**456 @ 4.3GB 7.2Krpm 624 @ 9.1GB 7.2Krpm 216 @ 9.1 GB 10Krpm**	**1.14 sec**	**92,832.96**

24 Clients: Hewlett-Packard Models C200 (8), C240 (14), C360 (2)		
CPU	**Memory**	**Disks**
PA-8200@ 200 MHz (C200) PA-8200@ 240 MHz (C240) PA-8500@ 367 MHz (C360)	**Main: 1.5 GB iCache: 512KB dCache: 1MB**	**1 @ 4.3 GB**

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

- **The transactions were correctly implemented.**
- **The database files were properly sized and populated.**
- **The database was properly scaled with 7,488 warehouses.**
- **I verified that additional disks on the measured system were not used in the measurement.**
- **The ACID properties were met.**
- **The ACID tests were performed on the measured database.**
- **Input data was generated according to the specified percentages.**
- **Eight hours of mirrored log space was present on the tested system.**
- **Space for eight hours of growth in dynamic tables was present on the tested system.**
- **The data for the 180-day space calculation was verified. Several adjustments were made in the priced configuration – see Auditor Notes. I verified that the replacements would not degrade the measurement.**
- **The steady state portion of the test was 26 minutes.**
- **One checkpoint was taken before the measured interval.**
- **One checkpoint was taken during the measured interval.**
- **The checkpoints were verified to be clear of the guard zone.**
- **The system pricing was checked for major components and maintenance.**

Auditor Notes:

The disks were attached to the disk controllers in 12-disk arrays with each array having the same sized disks. In the priced configuration, 18 of the 4.3GB disk arrays were changed to 9.1GB disk arrays. Two arrays of 4.3GB were not used in the measurement and were deleted in the priced configuration.

There were 8 SCSI disks directly connected to the system. Some were unused; others held OS Swap, the OS and DBMS. The space used by these components was included in the 180-day space calculation and taken from the storage arrays.

Three of the 10 arrays used for the log had 10Krpm drives; the remaining 7 had 7.2Krpm drives. The stripping was such that the pattern would remain the same over 8 hours.

The priced client machines all used the fastest processor.

Sincerely,

Tom Sawyer
Auditor

10 Report Availability

Requests for this TPC Benchmark C Full Disclosure Report should be sent to:

Transaction Processing
Performance Council
c/o Shanley Public Relations
777 North First Street
Suite 600
San Jose, CA 95112-6311

or your local Hewlett-Packard sales office.

Appendix A Client/Server Source

This appendix contains the source and makefiles for all client and server programs.

A.1 Transaction Source

client/buildenv.mk

```
OPTIMIZE= +O4 -z +Olibcalls +Ofastaccess +Oentrysched +Onolimit +Oprocelim +ESlit
+Xk

ARCHFLAGS= -Ae +DA2.0 +DS2.0 -DHPUX ## -D_REENTRANT

CINCLUDES= -I. -I../lib

BUILDFLAGS= $(ARCHFLAGS) $(OPTIMIZE) $(CINCLUDES)

MV=cp -r
```

client/client.c

```
/******************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 10:53:26 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
******************************************************************************/
#include "iobuf.h"
#include "tpcc.h"
#include <signal.h>

#define until(c)  while(!(c))


/* a generic transaction variable. */
generic_trans generic_transaction;
generic_trans *trans=&generic_transaction;

/* global variables set up during initialization */
int user;
ID warehouse;
ID district;

main(argc, argv)
    int argc;
    char **argv;
    {
    int key;

    /* setup the transactions */
    key = setup(argc, argv);

    /* repeat until done */
    while (key != '9' && key != EOF)
       {

        /* get the menu choice */
        key = menu_read();

        /* process according to the choice */
        switch(key)
            {
            case '1': key = neworder(&trans->neworder); break;
            case '2': key = payment(&trans->payment); break;
            case '3': key = ordstat(&trans->ordstat); break;
            case '4': key = delivery(&trans->delivery); break;
            case '5': key = stocklev(&trans->stocklev); break;
            case EOF: break;
            case '9': break;
            default:    msgline("Please enter a valid menu choice");
            }
       }

    /* done */
    cleanup();
    }


/******************************************************************************
******************************************************************************

Neworder form processing

******************************************************************************
******************************************************************************/

define_iobuf(neworder_form, 900);

int neworder(trans)
    neworder_trans *trans;
    {
    int key;
    display(neworder_form);
    key = neworder_read(trans);
    if (key != ENTER) return key;
    neworder_transaction(trans);
    neworder_write(trans);
    return key;
    }


int neworder_read(trans)
    neworder_trans *trans;
    {
    int i;
    int field;
    int key;
    int ol;

    /* Our warehouse number is fixed */
    trans->W_ID = warehouse;
    trans->D_ID = EMPTY_NUM;

    /* assume nothing set yet */
    trans->C_ID = EMPTY_NUM;
    for (i=0; i<15; i++)
        {
        trans->item[i].OL_I_ID = EMPTY_NUM;
        trans->item[i].OL_QUANTITY = EMPTY_NUM;
        trans->item[i].OL_SUPPLY_W_ID = EMPTY_NUM;
        }

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 47))
        retry: switch (field)
            {

            case 1: key = read_number(4, 29, &trans->D_ID, 2);
                    break;

            case 2: key = read_number(5, 12, &trans->C_ID, 4);
                    break;

            case 3: case 6: case 9: case 12: case 15:
            case 18: case 21: case 24: case 27: case 30:
            case 33: case 36: case 39: case 42: case 45:
                ol = (field - 3) / 3;
                key = read_number(9+ol, 3, &trans->item[ol].OL_SUPPLY_W_ID,4);
                break;
```

```
            case 4: case 7: case 10: case 13: case 16:
            case 19: case 22: case 25: case 28: case 31:
            case 34: case 37: case 40: case 43: case 46:
                ol = (field - 3) / 3;
                key = read_number(9+ol,10, &trans->item[ol].OL_I_ID, 6);
                break;

            case 5: case 8: case 11: case 14: case 17:
            case 20: case 23: case 26: case 29: case 32:
            case 35: case 38: case 41: case 44: case 47:
                ol = (field - 3) / 3;
                key = read_number(9+ol, 45, &trans->item[ol].OL_QUANTITY, 2);
                break;
            }

    /* abort the screen if requested */
    if (key != ENTER)
        return key;

    /* calculate how many items were entered */
    for (i=15; i>0; i--)
        if ((trans->item[i-1].OL_I_ID != EMPTY_NUM) ||
             (trans->item[i-1].OL_SUPPLY_W_ID != EMPTY_NUM) ||
             (trans->item[i-1].OL_QUANTITY != EMPTY_NUM)) break;
    trans->O_OL_CNT = i;

    /* make sure all necessary fields are filled in */
    if (trans->D_ID == EMPTY_NUM)
        {field=1; msgline("Please specify district"); goto retry;}
    if (trans->C_ID == EMPTY_NUM)
        {field=2; msgline("Please specify customer id"); goto retry;}
    if (trans->O_OL_CNT == 0)
        {field=3; msgline("Please enter at least one orderline"); goto retry;}
    for (i=0; i<trans->O_OL_CNT; i++)
        {
        if (trans->item[i].OL_SUPPLY_W_ID == EMPTY_NUM)
            {field=i*3+3; msgline("Please enter supply warehouse"); goto retry;}
        if (trans->item[i].OL_I_ID == EMPTY_NUM)
            {field=i*3+4; msgline("Please enter Item id"); goto retry;}
        if (trans->item[i].OL_QUANTITY == EMPTY_NUM
         || trans->item[i].OL_QUANTITY <= 0)
            {field=i*3+5; msgline("Please enter quantity > 0"); goto retry;}
        }

    /* decide if they were all local */
    for (i=0; i<trans->O_OL_CNT; i++)
        if (trans->item[i].OL_SUPPLY_W_ID != trans->W_ID) break;
    trans->all_local = (i == trans->O_OL_CNT);

    /* display number of order lines */
    number(6, 42, trans->O_OL_CNT, 2);

    msgline("");
    flush();
    return key;
    }



neworder_write(t)
    neworder_trans *t;
    {
    int i;
    MONEY amount, total_amount, cost;

    /* Rev. 3.3 error checking: both of the following branches are
     * skipped.  We'll go to status and print an error message.
     */

    /* CASE: invalid item, display only these values */
    if (t->status == E_INVALID_ITEM)
        {
        text(5, 25, t->C_LAST);
        text(5,52,  t->C_CREDIT);
        number(6, 15, t->O_ID, 8);
        }

    /* CASE: everything OK, display everything */
    else if (t->status == OK)
        {
        text(5, 25, t->C_LAST);
        text(5,52,  t->C_CREDIT);
        number(6, 15, t->O_ID, 8);
        date(4, 61, t->O_ENTRY_D);
        real(5, 64, t->C_DISCOUNT * 100, 5, 2);
        real(6, 59, t->W_TAX*100, 5, 2);
        real(6, 74, t->D_TAX*100, 5, 2);

        total_amount = 0;
        for (i=0; i < t->O_OL_CNT; i++)
            {

            /* keep track of amount of each line and total */
            amount = t->item[i].I_PRICE * t->item[i].OL_QUANTITY;
            total_amount += amount;

            /* display the item line */
            text(9+i, 19, t->item[i].I_NAME);
            number(9+i, 51, t->item[i].S_QUANTITY, 3);
            position(9+i, 58);   pushc(t->item[i].brand_generic);
            money(9+i, 62, t->item[i].I_PRICE, 7);
            money(9+i, 71, amount, 8);
            }

        /* Clear the screen of any empty input fields */
        clear_screen();

        /* display the total cost */
        text(24, 63, "Total:");
        cost = total_amount * (1 - t->C_DISCOUNT) * (1 + t->W_TAX + t->D_TAX);
        money(24, 71, cost, 9);
        }

    /* display the status message */
    status(24, 1, t->status);
    }


neworder_setup()
    {
    int item;
    iobuf *old;

    /* start with an empty form */
    reset(neworder_form);

    /* redirect the data to a special menu buffer */
    old = out_buf;  out_buf = neworder_form;

    /* clear the iobuf below the menu */
    position(3,1);
    clear_screen();

    /* set up all the field labels */
    text(3, 36, "New Order");
    text(4, 1, "Warehouse:");
    number(4, 12, warehouse, 4);
    text(4, 19, "District:");
    empty(4, 29, 2);
    text(4, 55, "Date:");
    text(5, 1, "Customer:");
    empty(5, 12, 4);
    text(5, 19, "Name:");
    text(5, 44, "Credit:");
    text(5, 57, "Disc.:");
    text(6, 1, "Order Number:");
    text(6, 25, "Number of Lines:");
    text(6, 52, "W_Tax:");
    text(6, 67, "D_Tax:");
    text(8, 2, "Supp_W Item_Num   Item_Name");
    text(8, 45, "Qty Stock   B/G     Price  Amount");

    /* display blank fields for each item */
    for (item = 1; item <= 15; item++)
        {
        empty(8+item, 3, 4);
        empty(8+item, 10, 6);
```

```
        empty(8+item, 45, 2);
        }

    trigger();

    /* restore to the previous I/O buffer */
    out_buf = old;
    }



/*************************************************************************
*************************************************************************

Payment form processing

*************************************************************************
*************************************************************************/

define_iobuf(payment_form, 400);


int payment(trans)
    payment_trans *trans;
    {
    int key;
    display(payment_form);
    key = payment_read(trans);
    if (key != ENTER)  return key;
    payment_transaction(trans);
    payment_write(trans);
    return key;
    }


payment_setup()
    {
    int item;
    iobuf *old;

    /* start with an empty form */
    reset(payment_form);

    /* redirect the data to a special menu buffer */
    old = out_buf;  out_buf = payment_form;

    /* clear the iobuf below the menu */
    position(3,1);
    clear_screen();

    /* set up all the field labels */
    text(3, 38, "Payment");
    text(4, 1, "Date:");
    text(6, 1, "Warehouse:");
    number(6, 12, warehouse, 4);
    text(6, 42, "District:");
    empty(6, 52, 2);
    text(11, 1, "Customer:");
    empty(11, 11, 4);
    text(11, 17, "Cust-Warehouse:");
    empty(11, 33, 4);
    text(11, 39, "Cust-District:");
    empty(11, 54, 2);
    text(12, 1, "Name:");
    empty(12, 29, 16);
    text(12, 50, "Since:");
    text(13, 50, "Credit:");
    text(14, 50, "%Disc:");
    text(15, 50, "Phone:");
    text(17, 1, "Amount Paid:");
    empty(17, 23, 8);
    text(17, 37, "New Cust-Balance:");
    text(18, 1, "Credit Limit:");
    text(20, 1, "Cust-Data:");
    trigger();

    out_buf = old;
    }


int payment_read(t)
    payment_trans *t;
    {
    int i;
    int field;
    int key;

    /* Our warehouse number is fixed */
    t->W_ID = warehouse;
    t->C_ID = EMPTY_NUM;
    t->D_ID = EMPTY_NUM;
    t->C_W_ID = EMPTY_NUM;
    t->C_D_ID = EMPTY_NUM;
    t->H_AMOUNT = EMPTY_FLT;
    t->C_LAST[0] = '\0';

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 6))
        retry: switch (field)
            {

            case 1: key = read_number(6, 52, &t->D_ID, 2);
                    break;

            case 2:
                /* if last name specified, skip this field */
                if (t->C_LAST[0] != '\0')
                    break;

                /* read in the customer id */
                key = read_number(11, 11, &t->C_ID, 4);

                /* if specified, don't allow last name to be entered */
                if (t->C_ID != EMPTY_NUM)
                    {
                    blanks(12, 29, 16);
                    t->C_LAST[0] = '\0';
                    }

                /* refresh the C_LAST underlines, if possibly needed */
                else if (t->C_LAST[0] == '\0')
                    empty(12, 29, 16);
                break;

            case 3: key = read_number(11, 33,  &t->C_W_ID, 4);
                    break;

            case 4: key = read_number(11, 54, &t->C_D_ID, 2);
                    break;

            case 5:
                /* skip this field if C_ID was already specified */
                if (t->C_ID != EMPTY_NUM)
                    break;

                /* read in the customer last name */
                key = read_text(12, 29, t->C_LAST, 16);

                /* if specified, don't allow c_id to be entered */
                if (t->C_LAST[0] != '\0')
                    {
                    blanks(11, 11, 4);
                    t->C_ID = EMPTY_NUM;
                    }

                /* refresh the C_ID underlines, if possibly needed */
                else if (t->C_ID == EMPTY_NUM)
                    empty(11, 11, 4);
                break;

            case 6: key = read_money(17, 23, &t->H_AMOUNT, 8);
                    break;
            }

        /* if Aborted, then done */
```

```
    if (key != ENTER)
        return key;


    /* Make sure all the fields were entered */
    if (t->D_ID == EMPTY_NUM)
        {field=1; msgline("Please enter district id"); goto retry;}
    if (t->C_ID == EMPTY_NUM && t->C_LAST[0] == '\0')
        {field=2; msgline("C_ID or C_LAST must be entered"); goto retry;}
    if (t->C_W_ID == EMPTY_NUM)
        {field=3; msgline("Please enter customer's warehouse"); goto retry;}
    if (t->C_D_ID == EMPTY_NUM)
        {field=4; msgline("please enter customer's district"); goto retry;}
    if (t->H_AMOUNT == EMPTY_FLT)
        {field=6; msgline("Please enter payment amount"); goto retry;}
    if (t->H_AMOUNT <= 0)
          {field=6; msgline("Please enter a positive payment"); goto retry;}


    t->byname = (t->C_ID == EMPTY_NUM);
    msgline("");
    flush();
    return key;
    }


payment_write(t)
    payment_trans *t;
    {

    /* if errors, display a message and quit */
    if (t->status != OK)
        {
        status(24, 1, t->status);
        return;
        }

    /* display the screen */
    date(4, 7, t->H_DATE);
    text(7, 1, t->W_STREET_1);
    text(7, 42, t->D_STREET_1);
    text(8, 1, t->W_STREET_2);
    text(8, 42, t->D_STREET_2);
    text(9, 1, t->W_CITY);
    text(9, 22, t->W_STATE);
    zip(9, 25, t->W_ZIP);
    text(9, 42, t->D_CITY);
    text(9, 63, t->D_STATE);
    zip(9, 66, t->D_ZIP);
    number(11, 11, t->C_ID, 4);
    text(12, 9, t->C_FIRST);
    text(12, 26, t->C_MIDDLE);
    text(12, 29, t->C_LAST);
    date_only(12, 58, t->C_SINCE);
    text(13, 9, t->C_STREET_1);
    text(13, 58, t->C_CREDIT);
    text(14, 9, t->C_STREET_2);
    real(14, 58, t->C_DISCOUNT*100, 5, 2);  /* percentage or fraction? */
    text(15, 9, t->C_CITY);
    text(15, 30, t->C_STATE);
    zip(15, 33, t->C_ZIP);
    phone(15, 58, t->C_PHONE);
    money(17, 17, t->H_AMOUNT,14);
    money(17, 55, t->C_BALANCE, 15);
    money(18, 17, t->C_CREDIT_LIM, 14);

    /* Display cust data if bad credit. */
    if (t->C_CREDIT[0] == 'B' && t->C_CREDIT[1] == 'C')
        long_text(20, 12, t->C_DATA, 50);
    }



/*****************************************************************
*****************************************************************

ORDSTAT form processing
*****************************************************************
*****************************************************************/

define_iobuf(ordstat_form, 300);



int ordstat(t)
    ordstat_trans *t;
    {
    int key;
    display(ordstat_form);
    key = ordstat_read(trans);
    if (key != ENTER)  return key;
    ordstat_transaction(trans);
    ordstat_write(trans);
    return key;
    }


ordstat_setup()
    {
    int item;
    iobuf *old;

    /* start with an empty form */
    reset(ordstat_form);

    /* redirect the data to a special menu buffer */
    old = out_buf;  out_buf = ordstat_form;

    /* clear the iobuf below the menu */
    position(3,1);
    clear_screen();

    /* set up all the field labels */
    text(3, 35, "Order-Status");
    text(4, 1, "Warehouse:");
    number(4, 12, warehouse, 4);
    text(4, 19, "District:");
    empty(4, 29, 2);
    text(5, 1, "Customer:");
    empty(5, 11, 4);
    text(5, 18, "Name:");
    empty(5, 44, 16);
    text(6, 1, "Cust-Balance:");
    text(8, 1, "Order-Number");
    text(8, 26, "Entry-Date:");
    text(8, 60, "Carrier-Number:");
    text(9, 1, "Supply-W");
    text(9, 14, "Item-Num");
    text(9, 25, "Qty");
    text(9, 33,"Amount");
    text(9, 45, "Delivery-Date");

    trigger();

    /* done */
    out_buf = old;
    }



int ordstat_read(t)
    ordstat_trans *t;
    {
    int i;
    int field;
    int key;

    /* Our warehouse number is fixed */
    t->W_ID = warehouse;
    t->C_ID = EMPTY_NUM;
    t->D_ID = EMPTY_NUM;
    t->C_LAST[0] = '\0';

    /* Process fields until done */
```

```
    for (field = 1; field > 0; field = next_field(field, key, 3))
        retry: switch (field)
            {
            case 1: key = read_number(4, 29, &t->D_ID, 2);
                    break;

            case 2:
                /* if last name specified, skip this field */
                if (t->C_LAST[0] != '\0')
                    break;

                /* read in the customer id */
                key = read_number(5, 11, &t->C_ID, 4);

                /* if specified, don't allow last name to be entered */
                if (t->C_ID != EMPTY_NUM)
                    {
                    blanks(5, 44, 16);
                    t->C_LAST[0] = '\0';
                    }

                /* refresh the C_LAST underlines, if possibly needed */
                else if (t->C_LAST[0] == '\0')
                    empty(5, 44, 16);
                break;

            case 3:
                /* skip this field if C_ID was already specified */
                if (t->C_ID != EMPTY_NUM)
                    break;

                /* read in the customer last name */
                key = read_text(5, 44, t->C_LAST, 16);

                /* if specified, don't allow c_id to be entered */
                if (t->C_LAST[0] != '\0')
                    {
                    blanks(5, 11, 4);
                    t->C_ID = EMPTY_NUM;
                    }

                /* refresh the C_ID underlines, if possibly needed */
                else if (t->C_ID == EMPTY_NUM)
                    empty(5, 11, 4);
                break;
            }

    /* if Aborted, then done */
    if (key != ENTER)
        return key;


    /* ensure all the necessary fields were entered */
    if (t->D_ID == EMPTY_NUM)
        {field=1; msgline("Please enter district id"); goto retry;}
    if (t->C_ID == EMPTY_NUM && t->C_LAST[0] == '\0')
        {field=2;  msgline("C_ID or C_LAST must be entered"); goto retry;}


    t->byname = (t->C_ID == EMPTY_NUM);
    msgline("");
    flush();
    return key;
    }


ordstat_write(t)
    ordstat_trans *t;
    {
    int i;

    /* if errors, display a status message and quit */
    if (t->status != OK)
        {
        status(24, 1, t->status);
        return;
        }

    /* display the results */
    number(5, 11, t->C_ID, 4);
    text(5, 24, t->C_FIRST);
    text(5, 41, t->C_MIDDLE);
    text(5, 44, t->C_LAST);
    money(6, 15, t->C_BALANCE, 10);
    number(8, 15, t->O_ID, 8);
    date(8, 38, t->O_ENTRY_DATE);
    if (t->O_CARRIER_ID > 0)
        number(8, 76, t->O_CARRIER_ID, 2);

    for (i=0; i< t->ol_cnt; i++)
        {
        number(i+10, 3, t->item[i].OL_SUPPLY_W_ID, 4);
        number(i+10, 14, t->item[i].OL_I_ID, 6);
        number(i+10, 25, t->item[i].OL_QUANTITY, 2);
        money(i+10, 32, t->item[i].OL_AMOUNT, 9);
        date_only(i+10, 47, t->item[i].OL_DELIVERY_DATE);
        }
    }




/*************************************************************
*************************************************************

delivery form processing

*************************************************************
*************************************************************/

define_iobuf(delivery_form, 300);



int delivery(t)
    delivery_trans *t;
    {
    int key;
    display(delivery_form);
    key = delivery_read(trans);
    if (key != ENTER)  return key;
    delivery_enque(trans);
    delivery_write(trans);
    return key;
    }


delivery_setup()
    {
    int item;
    iobuf *old;

    /* start with an empty form */
    reset(delivery_form);

    /* redirect the data to a special menu buffer */
    old = out_buf;  out_buf = delivery_form;

    /* clear the iobuf below the menu */
    position(3,1);
    clear_screen();

    /* set up all the field labels */
    text(3, 38, "Delivery");
    text(4, 1, "Warehouse:");
    number(4, 12, warehouse, 4);
    text(6, 1, "Carrier Number:");
    empty(6, 17, 2);

    trigger();

    /* done */
    out_buf = old;
    }
```

```
int delivery_read(t)
    delivery_trans *t;
    {
    int i;
    int field;
    int key;

    /* Our warehouse number is fixed */
    t->W_ID = warehouse;
    t->O_CARRIER_ID = EMPTY_NUM;

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 1))
        retry: switch (field)
            {
            case 1: key = read_number(6, 17, &t->O_CARRIER_ID, 2);
                    break;
            }

    /* if Aborted, then done */
    if (key != ENTER)
        return key;

    /* Must enter the carrier id */
    if ((t->O_CARRIER_ID == EMPTY_NUM) ||
          (t->O_CARRIER_ID < 1) ||
          (t->O_CARRIER_ID > 10))
        {field=1; msgline("Please enter a Carrier Number within 1 and 10"); goto
retry; }

    /* clear the message line */
    msgline("");
    flush();
    return key;
    }


delivery_write(t)
    delivery_trans *t;
    {
    if (t->status == OK)
        text(8, 1, "Execution Status: Delivery has been queued");
    else
        status(8, 1, t->status);
    }



/*****************************************************************
*****************************************************************

stocklev form processing

*****************************************************************
*****************************************************************/

define_iobuf(stocklev_form, 300);



int stocklev(t)
    stocklev_trans *t;
    {
    int key;
    display(stocklev_form);
    key = stocklev_read(trans);
    if (key != ENTER)  return key;
    stocklev_transaction(trans);
    stocklev_write(trans);
    return key;
    }


stocklev_setup()
    {
    int item;
    iobuf *old;

    /* start with an empty form */
    reset(stocklev_form);

    /* redirect the data to a special menu buffer */
    old = out_buf;  out_buf = stocklev_form;

    /* clear the iobuf below the menu */
    position(3,1);
    clear_screen();

    /* set up all the field labels */
    text(3, 35, "Stock-Level");
    text(4, 1, "Warehouse:");
    number(4, 12, warehouse, 4);
    text(4, 19, "District:");
    number(4, 29, district, 2);
    text(6, 1, "Stock Level Threshold:");
    empty(6, 24, 2);
    text(8, 1, "low stock");

    trigger();

    /* done */
    out_buf = old;
    }



int stocklev_read(t)
    stocklev_trans *t;
    {
    int field;
    int key;

    t->W_ID = warehouse;
    t->D_ID = district;
    t->threshold = EMPTY_NUM;

    /* Process fields until done */
    for (field = 1; field > 0; field = next_field(field, key, 1))
        retry: switch (field)
            {
            case 1: key = read_number(6, 24, &t->threshold, 2);
                    break;
            }

    /* if Aborted, then done */
    if (key != ENTER)
        return key;


    /* make sure the necessary fields were entered */
    if ((t->threshold == EMPTY_NUM) ||
        (t->threshold < 10) ||
          (t->threshold > 20))

        {field=1; msgline("Please enter a threshold within 10 and 20"); goto retry;
}

    /* clear the message line */
    msgline("");
    flush();
    return key;
    }


stocklev_write(t)
    stocklev_trans *t;
    {
    if (t->status == OK)
        number(8, 12, t->low_stock, 3);
    else
       status(10, 1, t->status);
    }
```

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

login form processing

**********************************************************************
*********************************************************************/

int login()
    {
    int field;
    int key;
    char auditstr[21];
    int w_id, d_id;

    /* assume the default values */
    w_id = warehouse;
    d_id = district;
    auditstr[0] = '\0';

    /* display the login menu */
    position(1,1);  clear_screen();
    text(3, 30, "Please login.");
    text(5,5,"Warehouse:");
    number(5, 16, w_id, 4);
    text(5, 24, "District:");
    number(5, 34, d_id, 2);
    text(15, 5, "Audit String:");
    text(15, 19, CLIENT_AUDIT_STRING);
    empty(16, 19, 20);
    trigger();

    /* Get values until done */
    for (field = 1; field > 0;  field = next_field(field, key, 3))
        retry: switch (field)
            {
            case 1:
                key = read_number(5, 16, &w_id, 4, Num);
                break;

            case 2:
                key = read_number(5, 34, &d_id, 2, Num);
                break;

            case 3:
                key = read_text(16, 19, auditstr, 20);
                break;
            }

    if (key != ENTER)
        return EOF;

    if (w_id == EMPTY_NUM && warehouse == EMPTY_NUM)
        {
        msgline("You must enter a warehouse id");
        field =1;
        goto retry;
        }

    if (d_id == EMPTY_NUM && district == EMPTY_NUM)
        {
        msgline("You must enter a district id");
        field = 2;
        goto retry;
        }

    if (w_id != EMPTY_NUM)
        warehouse = w_id;
    if (d_id != EMPTY_NUM)
        district = d_id;

    /* done */
    flush();
    return key;
    }
```

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

menu form processing

**********************************************************************
*********************************************************************/

menu_setup()
    {

    /* display the menu on the iobuf -- never erased */
    position(1, 1);
    clear_screen();
    string("(1)New-Order (2)Payment (3)Order-Status ");
    string("(4)Delivery (5)StockLevel (9)Exit");
    }

int menu_read()
    {
    position(1, 1);
    trigger();
    return getkey();
    }



int next_field(current, key, max)
    int current;
    int key;
    int max;
    {

    if (key == BACKTAB)
        if (current == 1)    return max;
        else                 return current-1;
    else if (key == TAB)
        if (current == max)  return 1;
        else                 return current+1;
    else
        return 0;
    }



msgline(str)
    char *str;
    {
    position(24, 1);
    clear_screen();
    string(str);
    flush();  /* Needed? */
    }



int setup(argc, argv)
    int argc;
    char **argv;
    {
    int key;

    /* Ignore SIGPIPE, since they occur normally */
    signal(SIGPIPE, SIG_IGN);

    /* get the user, warehouse and district numbers */
    warehouse = EMPTY_NUM;
    district = EMPTY_NUM;
    key = login();
    user = warehouse*DIST_PER_WARE + district + 1;
```

```
    /* set up the forms */
    menu_setup();
    neworder_setup();
    payment_setup();
    ordstat_setup();
    delivery_setup();
    stocklev_setup();

    /* connect to the delivery queue */
    delivery_init(user);

    /* connect to the transaction processor */
    transaction_begin(user);

    return key;
    }


cleanup()
    {

    /* detach from transaction engine */
    transaction_done();

    /* detach from the delivery queue */
    delivery_done();

    /* clear the screen */
    position(1, 1);
    clear_screen();
    flush();
    }


/*************************************************************************
**************************************************************************

Screen Output Routines

**************************************************************************
**************************************************************************/


number(row, col, n, width)
    int row;
    int col;
    int n;
    int width;
    {
    char str[81];
    fmt_num(str, n, width);
    text(row, col, str);
    }



real(row, col, x, width, dec)
    int row;
    int col;
    double x;
    int width;
    int dec;
    {
    char str[81];
    fmt_flt(str, x, width, dec);
    text(row, col, str);
    }

date(row, col, date_str)
    int row;
    int col;
    char *date_str;
    {
    text(row, col, date_str);
    }
```

```
date_only(row, col, date_str)
    int row;
    int col;
    char *date_str;
    {
    date_str[10] = '\0';
    text(row, col, date_str);
    }

money(row, col, x, width)
    int row;
    int col;
    double x;
    int width;
    {
    char str[81];
    fmt_money(str, x, width);
    text(row, col, str);
    }


long_text(row, col, str, width)
    int row, col, width;
    char *str;
    {
    int pos;

    /* repeat until the entire string is written out */
    for (pos = width; *str != '\0';  str++, pos++)
        {

        /* if at end of line, position the cursor to next line */
        if (pos >= width)
            {
            position(row, col);
            pos = 0;
            row++;
            }

        /* output the next character */
        pushc(*str);
        }
    }


text(row, col, str)
    int row;
    int col;
    char str[];
    {
    position(row, col);
    string(str);
    }


phone(row, col,str)
      int row;
      int col;
      char *str;
      {

       char temp[30];

       fmt_phone(temp,str);
       text(row,col,temp);

       }


zip(row, col, str)
    int row;
    int col;
    char *str;
    {
    char temp[30];

    fmt_zip(temp,str);
    text(row,col,temp);
    }
```

```
empty(row, col, len)
    int row;
    int col;
    int len;
    {
    position(row, col);
    while (len-- > 0)
        pushc('_');
    }


blanks(row, col, len)
    int row, col, len;
    {
    position(row, col);
    while (len-- > 0)
        pushc(' ');
    }


status(row, col, status)
/*****************************************************************
status displays the transaction status
  Note: must correspond to 'get_status' in driver/keystroke.c
*****************************************************************/
   int row, col;
   int status;
   {
   text(row, col, "Execution Status: ");

   if (status == OK)
       string("Transaction Committed");
   else if (status == E_INVALID_ITEM)
       string("Item number is not valid");
   /* Do the rev. 3.3 error checking here.        */
   else if (status == E_INVALID_INPUT)
       string("Invalid input, transaction not executed");
   else
       {
       string("Rollback -- ");
       number(row, col+30, status, 5);
       }
   }

/*****************************************************************
******************************************************************

ASCII terminal control

******************************************************************
******************************************************************/


trigger()
/*****************************************************************
trigger sends a turnaround sequence to let the driver know to send input
*****************************************************************/
    {
    pushc(TRIGGER);
    }


position(row, col)
/*****************************************************************
position positions the cursor at the given row and column
*****************************************************************/
    int row;
    int col;
    {
    pushc(ESCAPE);
    pushc('[');
    if (row >= 10)
        pushc('0' + row/10);
    pushc('0'+ row%10);
    pushc(';');
    if (col >= 10)
        pushc('0' + col/10);
    pushc('0' + col%10);
    pushc('H');
    }

clear_screen()
/*****************************************************************
clear_screen clears the iobuf from cursor position to end of iobuf
*****************************************************************/
    {
    pushc(ESCAPE);
    pushc('[');
    pushc('J');
    }




/*****************************************************************
******************************************************************

Screen Input Routines

******************************************************************
******************************************************************/
#define funny(key)  (key != ENTER && key !=TAB && key != BACKTAB)


read_number(row, col, n, width)
/*****************************************************************
read_number reads an integer field
*****************************************************************/
    int row;
    int col;
    int *n;
    int width;
    {
    char temp[81];
    int key;
    int err;
    debug("read_number: row=%d  col=%d  width=%d n=%d \n",row, col,width,*n);

    /* generate the current characters */
    fmt_num(temp, *n, width);
    err = NO;

    /* repeat until a valid number or a funny key is pressed */
    for (;;)
        {
        /* Let the user edit the field */
        key = getfield(row, col, temp, width, Num);
          if (funny(key)) return key;

        /* convert the field to a number */
        *n = cvt_num(temp);
        if (*n != INVALID_NUM) break;

        msgline("Invalid digit entered");
          pushc(BELL);
        err = YES;
        }

    /* display the new number */
    number(row, col, *n, width);
    if (err) msgline("");
    debug("read_number: n=%d  key=%d\n", *n, key);
    return key;
    }


int read_money(row, col, m, width)
    int row;
    int col;
    double *m;
    int width;
    {
    char temp[81];
    int key;
    int err;
```

```
    err = NO;
    fmt_money(temp, *m, width);

    /* repeat until a valid number or a funny key is pressed */
    for (;;)
        {
        key = getfield(row, col, temp, width, Money);
          if (funny(key)) return key;

        *m = cvt_money(temp);
        if (*m != INVALID_FLT)  break;

        msgline("Please enter amount $99999.99");
        pushc(BELL);
        err = YES;
          }

    money(row, col, *m, width);
    if (err) msgline("");
    return key;
    }



int read_real(row, col, x, width, dec)
    int row, col, width;
    double *x;
    {
    char temp[81];
    int key;
    int err;

    /* generate the current characters */
    fmt_flt(temp, *x, width, dec);
    err = NO;

    /* repeat until a valid number or a funny key is pressed */
    for (;;)
          {
        key = getfield(row, col, temp, width);
          if (funny(key)) return key;

        /* convert the field to a number */
        *x = cvt_flt(temp);
          if (*x != INVALID_FLT) break;

        msgline("Please enter a valid floating pt number");
        pushc(BELL);
        err = YES;
        }

    /* display the new number */
    real(row, col, *x, width, dec);
    if (err) msgline("");

    return key;
    }



int read_text(row, col, s, width)
    int row, col, width;
    char *s;
    {
    char temp[81];
    int key;
    int i;

    /* generate the current characters */
    fmt_text(temp, s, width);

    /* let the user edit the field */
    key = getfield(row, col, temp, width, Text);
    if (funny(key)) return key;

    /* Strip off leading and trailing space characters */
    cvt_text(temp, s);

    /* redisplay the current text */
    fmt_text(temp, s, width);
    text(row, col, temp);

    return key;
    }



int getfield(row, col, buf, width,ftype)
    int row, col, width;
    char buf[];
    FIELD_TYPE ftype;

    {
    int pos, key;

    debug("getfield: width=%d  buf=%*s\n", width, width, buf);

    /* go to the beginning of the field */
    position(row, col);
    pos = 0;

    /* repeat until a special control character is pressed */
    for (;;)
        {

        /* get the next character */
        key = getkey();

        /* CASE: Add to buf if it fits and Is it a valid character ? */
        if (pos < width  && valid_char(key, ftype))
            {
            buf[pos] = key;
            pos++;
            pushc(key);
            }

        /* CASE: char is BACKSPACE.  Erase last character. */
        else if (key == BACKSPACE && pos > 0)
            {
            pos--;
            buf[pos] = '_';
            pushc(BACKSPACE);
            pushc('_');
            pushc(BACKSPACE);
            }

       /* CASE: enter, tab, backtab, ^c.  Exit loop */
       else if (key==ENTER || key==TAB || key==BACKTAB || key==CNTRLC
                || key == EOF)
           break;

       else if (key=='\031')   /* for debugging, let ^X == ENTER */
            {key=ENTER; break;}

       /* Otherwise, ignore the character and beep */
       else
           pushc(BELL);
       }

    debug("getfield: final key: %d  buf=%*s\n", key, width, buf);
    return key;
    }



int valid_char(key, ftype)
/*********************************************************************
valid_char is true if the key is valid for this type of field
*********************************************************************/
    int key;
    FIELD_TYPE ftype;
    {
    int valid;
    switch(ftype)
           {
           case Num : valid = (isdigit(key) || key == '-' || key == ' ');
```

```
                    break;

        case Text : valid = (isprint(key) || key == ' ');
                    break;

        case Money : valid = (isdigit(key) || key == '-' || key == '.'
                           || key == '$' || key == ' ');
                     break;

        default     : valid = NO;
                    break;
        }

     return valid;
     }
```

client/tux_transaction.c

```
/*******************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 10:53:27 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*******************************************************************************/

#include <varargs.h>
#include <errno.h>


#include "atmi.h"
#include "Uunix.h"
#include "tpcc.h"
int user;


neworder_trans *neworder_ptr;
payment_trans  *payment_ptr;
ordstat_trans *ordstat_ptr;
stocklev_trans *stocklev_ptr;
delivery_trans *delivery_ptr;

int result;



transaction_begin(u)
    int u;
    {

    /* keep track of which user we are (for error messages only) */
    user = u;

    /* attach to Tuxedo */
    if (tpinit( (TPINIT *)NULL) == -1)
        tux_error("Failed to attach to Tuxedo\n");

    /* allocate structures for each transaction */
    neworder_ptr = tpalloc("CARRAY", NULL, sizeof(neworder_trans));
    payment_ptr = tpalloc("CARRAY", NULL, sizeof(payment_trans));
    ordstat_ptr = tpalloc("CARRAY", NULL, sizeof(ordstat_trans));
    stocklev_ptr = tpalloc("CARRAY", NULL, sizeof(stocklev_trans));
    delivery_ptr = tpalloc("CARRAY", NULL, sizeof(delivery_trans));
    if (neworder_ptr == NULL || payment_ptr == NULL || ordstat_ptr == NULL
        || stocklev_ptr == NULL || delivery_ptr == NULL)
        tux_error("Unable to allocate Tuxedo memory\n");
    }



transaction_done()
    {
    if (tpterm() == -1)
        tux_error("Unable to detach from Tuxedo\n");
    }
```

```
void neworder_transaction(t)
    neworder_trans *t;
    {
    *neworder_ptr = *t;
    while (tpcall("NEWO_SVC", neworder_ptr, sizeof(neworder_trans),
              &neworder_ptr, &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for neworder transaction\n");
          *neworder_ptr = *t;
          }
    *t = *neworder_ptr;
    }



void payment_transaction(t)
    payment_trans *t;
    {
    *payment_ptr = *t;
    while (tpcall("PMT_SVC", payment_ptr, sizeof(payment_trans),
                          &payment_ptr, &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for payment transaction\n");
          *payment_ptr = *t;
          }
    *t = *payment_ptr;
    }


void ordstat_transaction(t)
    ordstat_trans *t;
    {
    *ordstat_ptr = *t;
    while (tpcall("ORDS_SVC", ordstat_ptr, sizeof(ordstat_trans),
                           &ordstat_ptr, &result, TPSIGRSTRT|TPNOTIME) == -1) {
          tux_error("Tuxedo failed for ordstat transaction\n");
          *ordstat_ptr = *t;
          }
    *t = *ordstat_ptr;
    }



stocklev_transaction(t)
    stocklev_trans *t;
    {
    *stocklev_ptr = *t;
    while (tpcall("STKL_SVC", stocklev_ptr, sizeof(stocklev_trans),
                          &stocklev_ptr, &result, TPSIGRSTRT|TPNOTIME) == -1) {
        tux_error("Tuxedo failed for stocklev transaction\n");
          *stocklev_ptr = *t;
          }
    *t = *stocklev_ptr;
    }


delivery_init(u)
    int u;
    {
    }


delivery_enque(t)
    delivery_trans *t;
    {
    gettimeofday(&t->enque, NULL);
    t->status = OK;

    *delivery_ptr = *t;
    while (tpacall("DVRY_SVC", delivery_ptr, sizeof(delivery_trans),
                TPNOREPLY) == -1) {
        tux_error("Tuxedo failed enqueing delivery transaction\n");
          *delivery_ptr = *t;
          }
    }


delivery_done()
    {
    }
```

```
static tux_error(format, va_alist)
    char *format;
    va_dcl
    {
    va_list argptr;

    va_start(argptr);
    vmessage(format, argptr);

    message("Tuxedo error %d\n", tperrno);

    errno = Uunixerr;
    if (tperrno == TPEOS)
        syserror("Tuxedo encountered O/S error\n");

    if (tperrno != TPESVCERR)
          error("EXITING !!!\n");
    else
          message("Retrying transation\n");

    }
```

client/Makefile

```
#********************************************************************************
#@(#) Version: A.10.10 $Date: 98/02/03 08:13:40 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
#********************************************************************************

include ../buildenv.mk

#
# Makefile for compiling the client, batch-tpcc, and service code
#

OH      = ${ORACLE_HOME}
OL      = ${ORACLE_HOME}/lib
ORL     = ${ORACLE_HOME}/rdbms/lib
P       = ${WORK_DIR}/src
I       = $(P)/lib
L       = $(P)/lib
D       = $(P)/driver
Q       = $(P)/que
S       = $(P)/client

##include $(ORACLE_HOME)/bench/buildtools/prefix.mk

SH_OPT     = -Wl,-a,shared
OPT        =  -Wl,-a,archive_shared
LDOPTS     = -ldld -a archive_shared

TUXEDO     = -D_HPUX_SOURCE ${ROOTDIR}/include ${OPT}

ORA_CFLAGS = -DORA_BLOCK_PATH=\"/project/tpcc/blocks/\" \
             -DORA_NULL_DATE=\"01-01-1811\" \
             -D_HPUX_SOURCE -DSS_64BIT_SERVER -DHPPA64 -DSLS8NATIVE -DSLU8NATIVE \
             -DSLTS_ENABLE -DHPUX_KTHREAD -DSLXMX_ENABLE -DSLMXMX_ENABLE

ORA_INCLUDE=-I. -I${S}/oracle \
    -I$(ORACLE_HOME)/rdbms/demo \
    -I$(ORACLE_HOME)/rdbms/public -I$(ORACLE_HOME)/rdbms/include \
    -I$(ORACLE_HOME)/plsql/public \
    -I$(ORACLE_HOME)/network/public
SYB_INCLUDE= -I${SYBASE}/include
VIS_INCLUDE= -I ${VISIGENIC}/include
TUX_INCLUDE= -I${ROOTDIR}/include
INCLUDE    = -I. -I$L

SH_CFLAGS  = $(BUILDFLAGS) ${SH_OPT} ${INCLUDE} ${TUX_INCLUDE}
CFLAGS       = $(BUILDFLAGS) ${OPT} ${INCLUDE} ${TUX_INCLUDE}
CFLAGS_SYB = $(BUILDFLAGS) ${OPT} ${INCLUDE} ${TUX_INCLUDE} ${SYB_INCLUDE}
CFLAGS_ORA = $(BUILDFLAGS) ${ORA_CFLAGS} ${OPT} ${INCLUDE} ${ORA_INCLUDE}
${TUX_INCLUDE}
CFLAGS_SQL = -Dunix -D_HPUX_SOURCE -DVG_UNIX ${OPT} ${INCLUDE} ${TUX_INCLUDE}
${SQL_INCLUDE} ${VIS_INCLUDE}

ORA_LOAD =  -L${OL} -L${ORL} \
${ORL}/ssdbaed.o ${ORL}/kpudfo.o \
${OL}/nautab.o ${OL}/naeet.o ${OL}/naect.o ${OL}/naedhs.o \
-lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lnro8 -lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lclient8 -lvsn8 -lcommon8 -lskgxp8 -lgeneric8 \
-lmm \
-lnls8 -lcore8 -lnls8 -lcore8 -lnls8 \
-lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lnro8 -lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lclient8 -lvsn8 -lcommon8 -lskgxp8 -lgeneric8 \
-ltrace8 \
-lnls8 -lcore8 -lnls8 -lcore8 -lnls8 \
-lclient8 -lvsn8 -lcommon8 -lskgxp8 -lgeneric8 \
-lnls8 -lcore8 -lnls8 -lcore8 -lnls8 \
-lldapclnt8 \
-lcl -lm -lpthread -l:libc.sl -ldld

#`cat ${OL}/sysliblist`\
#

LDFLAGS_ORA= ${OPT} $(CFLAGS_ORA) ${L}/tpc_lib.a ${ORA_LOAD}
LDFLAGS_SYB= ${OPT} ${L}/tpc_lib.a -L${SYBASE}/lib -lsybdb -lm
LDFLAGS_SQL= ${OPT} ${L}/tpc_lib.a -L/opt/odbc/lib -lodbc -lm

PROGRAMS     = client service startup client_batch msg_server raw

install: others_oracle

all_ora: others_oracle service_oracle tpcc_client

tpcc_client: client
          $(MV) client ${WORK_DIR}/bin/
others_sybase: raw startup client_batch_syb msg_server_syb
          $(MV) raw startup client_batch msg_server ${WORK_DIR}/bin
others_oracle: raw startup client_batch_ora
          $(MV) raw startup client_batch ${WORK_DIR}/bin
others_sqlserver: raw startup client_batch_sql msg_server_sql
          $(MV) raw startup client_batch msg_server ${WORK_DIR}/bin
service_oracle: service_ora
          $(MV) service ${WORK_DIR}/bin/
service_sybase: service_syb
          $(MV) service ${WORK_DIR}/bin/
service_sqlserver: service_sql
          $(MV) service ${WORK_DIR}/bin/

${S}/sybase/transaction.o: ${S}/sybase/transaction.c
          $(CC) ${CFLAGS_SYB} $(L)/tpc_lib.a -c ${S}/sybase/transaction.c;
${S}/sqlserver/transactionb.o: ${S}/sqlserver/transactionb.c
          $(CC) ${CFLAGS_SQL} $(L)/tpc_lib.a -c ${S}/sqlserver/transactionb.c;

ORA_OBJS=plnew.o plord.o plpay.o pldel.o plsto.o tpccpl.o

transaction.o: ${S}/oracle/transaction.c
          $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/transaction.c;
plnew.o: ${S}/oracle/plnew.c
          $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plnew.c;
plord.o: ${S}/oracle/plord.c
          $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plord.c;
plpay.o: ${S}/oracle/plpay.c
          $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plpay.c;
pldel.o: ${S}/oracle/pldel.c
          $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/pldel.c;
plsto.o: ${S}/oracle/plsto.c
          $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plsto.c;
tpccpl.o: ${S}/oracle/tpccpl.c
```

```
	$(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/tpccpl.c;

raw: raw.o
	cc ${CFLAGS} raw.o $(L)/tpc_lib.a -o raw

startup: startup.o $(L)/tpc_lib.a
	cc ${SH_CFLAGS} startup.o $(L)/tpc_lib.a -o startup
	chmod a+rw startup

# Warning: can't use +pd because kernel will use this size to extend
# DATA pregion when break/sbreak is called.
# Force shared libc to avoid to both libc data from the archived and shared version.
client: client.o  tux_transaction.o $(L)/tpc_lib.a
	cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
	${ROOTDIR}/bin/buildclient -v -o client \
       -f "$(BUILDFLAGS) $(OPT) -Wl,+pi 256K -Wl,-R 0x100000 \
           client.o tux_transaction.o $(L)/tpc_lib.a" \
	-l "-lnsl  -lm -Wl,-ashared -lc"
	cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

service_ora: service.o transaction.o $(ORA_OBJS) $(L)/tpc_lib.a
	cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
	mv /project/iti/lib/libgp.a /project/iti/lib/libgp.a.cc_service
	${ROOTDIR}/bin/buildserver -v -b shm \
		-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC \
		-o service \
		-f "$(BUILDFLAGS) $(OPT) \
		    -Wl,+pi 4M -Wl,+pd 256K \
		    -Wl,-R 0x400000 -Wl,-D 0x40100000 \
		        service.o transaction.o $(ORA_OBJS) $L/tpc_lib.a \
		    ${LDFLAGS_ORA}" \
		-l "-lnsl "
	mv /project/iti/lib/libgp.a.cc_service /project/iti/lib/libgp.a
	cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

service_syb: service.o ${S}/sybase/transaction.o $(L)/tpc_lib.a
	cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
	${ROOTDIR}/bin/buildserver -v -b shm \
		-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC \
		-o service \
		-f "service.o transaction.o $L/tpc_lib.a \
		    ${SYBASE}/lib/libsybdb.a -lm";
	cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

service_sql: service.o ${S}/sqlserver/transactionb.o $(L)/tpc_lib.a
	cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
	${ROOTDIR}/bin/buildserver -v -b shm \
		-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC \
		-o service \
		-f "service.o transactionb.o $L/tpc_lib.a \
		   /vsbuild/v1.10/build/com/obj/inst/libodbc.sl"
	cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

client_batch_ora: $(D)/driver.o $(D)/generate.o transaction.o $(ORA_OBJS)
$(Q)/dummy_que.o $(L)/tpc_lib.a  \
		  $(L)/server_default.o
	$(CC) $(BUILDFLAGS) $(OPT) \
	-Wl,+pi 4M -Wl,+pd 256K \
	-Wl,-R 0x400000 -Wl,-D 0x40100000 \
	$(D)/driver.o $(D)/generate.o transaction.o \
	$(ORA_OBJS) $(Q)/dummy_que.o $(L)/server_default.o $(L)/tpc_lib.a \
	${LDFLAGS_ORA} -o client_batch;

client_batch_syb: $(D)/driver.o $(D)/generate.o transaction.o $(Q)/dummy_que.o
$(L)/tpc_lib.a  \
		  $(L)/server_default.o
	$(CC) $(D)/driver.o $(D)/generate.o transaction.o $(Q)/dummy_que.o
$(L)/server_default.o $(L)/tpc_lib.a ${LDFLAGS_SYB} -o client_batch;

client_batch_sql: $(D)/driver.o $(D)/generate.o transactionb.o $(Q)/dummy_que.o
$(L)/tpc_lib.a \
		                  $(L)/server_default.o
		  $(CC) $(D)/driver.o $(D)/generate.o transactionb.o
$(Q)/dummy_que.o $(L)/server_default.o $(L)/tpc_lib.a ${LDFLAGS_SQL} -o
client_batch;

msg_server_ora: $(Q)/msg_server.o  transaction.o $(ORA_OBJS) $(L)/tpc_lib.a
	$(CC) $(Q)/msg_server.o transaction.o $(ORA_OBJS) ${LDFLAGS_ORA} -o
msg_server;

msg_server_syb: $(Q)/msg_server.o  transaction.o $(L)/tpc_lib.a
	$(CC) $(Q)/msg_server.o transaction.o ${LDFLAGS_SYB} -o msg_server;

msg_server_sql: $(Q)/msg_server.o  transactionb.o $(L)/tpc_lib.a
	$(CC) $(Q)/msg_server.o transactionb.o ${LDFLAGS_SQL} -o msg_server;

clean:
	rm -f *.o

clobber: clean
	rm -f ${PROGRAMS}
```

client/service.c

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 10:53:26 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/

#include <unistd.h>
#include <sys/types.h>
#include "tpcc.h"
#include "atmi.h"

extern int userid;
char *cmd           = NULL;

int tpsvrinit(argc, argv)
int    argc;
char **argv;
{
    char c;
    int ret;

    /*
     * search for the options
     *    "-n" server number
     *    "-S" server program
     * purpose: to get svr_id & progname for DVRY_LOG files
     *
     */
    while ((c = getopt(argc, argv, "n:S:h:")) != EOF) {
        switch(c) {
        case 'n':
            userid = atoi(optarg);
            break;
        case 'S':
            cmd  = optarg;
            break;
       }
    }

    ret = transaction_begin(userid);
    results_open(userid);

    return 0;
}


void NEWO_SVC(svcinfo)
TPSVCINFO *svcinfo;
{
    neworder_transaction((neworder_trans *)svcinfo->data);
    tpreturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}


void PMT_SVC(svcinfo)
TPSVCINFO *svcinfo;
{

    payment_transaction((payment_trans *)svcinfo->data);
    tpreturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}
```

```
void ORDS_SVC(svcinfo)
TPSVCINFO *svcinfo;
{
    ordstat_transaction((ordstat_trans *)svcinfo->data);
    tpreturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

void STKL_SVC(svcinfo)
TPSVCINFO *svcinfo;
{
    stocklev_transaction((stocklev_trans *)svcinfo->data);
    tpreturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

void DVRY_SVC(svcinfo)
TPSVCINFO *svcinfo;
{
    delivery_trans *t = (delivery_trans *)svcinfo->data;
    gettimeofday(t->deque, NULL);
    delivery_transaction(t);
    gettimeofday(t->complete, NULL);
    results(t);

    /* Why do we return things ? */
    tpreturn(TPSUCCESS, 0, svcinfo->data, svcinfo->len, 0);
}

/*************************************************************
tpsrvdone cleans up after the TPC transaction service
*************************************************************/
void tpsvrdone()
{
    transaction_done();
    results_close();

    /* Log a message saying we are done */
    message("TUXEDO service %s has shutdown\n", cmd);
}
```

client/oracle/transaction.c

```
#include "ora_tpcc.h"
#include <time.h>
#include "tpcc.h"

#ifdef __STDC__
#include "ociapr.h"
#else
#include "ocikpr.h"
#endif

extern TPCinit(int, char*, char*);

int     numtrans = 0;

transaction_done ()
{
   TPCexit();
   message("TPCexit after %d transcations \n", numtrans);
}

/* void  */
transaction_begin(id)
int id;
{
   int ret;

   if ((ret=TPCinit(id, "tpcc", "tpcc")) == -1)
   {
         message("TPCinit failure!\n");
        /* Error */
   }
   numtrans = 0;
   return ret;
}

void neworder_transaction(str)
neworder_trans *str;
{
   int i;
   struct newstruct ora_str;

   ora_str.newin.w_id = str->W_ID;
   ora_str.newin.d_id = str->D_ID;
   ora_str.newin.c_id = str->C_ID;
   for (i = 0; i < str->O_OL_CNT; i++) {
      ora_str.newin.ol_i_id[i] = str->item[i].OL_I_ID;
      ora_str.newin.ol_supply_w_id[i] = str->item[i].OL_SUPPLY_W_ID;
      ora_str.newin.ol_quantity[i] = str->item[i].OL_QUANTITY;
   }
   for (i = str->O_OL_CNT; i < 15; i++) {
      ora_str.newin.ol_i_id[i] = 0;
      ora_str.newin.ol_supply_w_id[i] = 0;
      ora_str.newin.ol_quantity[i] = 0;
   }

   numtrans++;
   if (TPCnew(&ora_str) == -1) {
      str->status = E_DB_ERROR;
      return;
   } else {
      str->status = OK;
   }

   str->O_ID = ora_str.newout.o_id;
   str->O_OL_CNT = ora_str.newout.o_ol_cnt;
   strncpy (str->C_LAST, ora_str.newout.c_last, 17);
   strncpy (str->C_CREDIT, ora_str.newout.c_credit, 3);
   str->C_DISCOUNT = (REAL) ora_str.newout.c_discount;
   str->W_TAX = (REAL) ora_str.newout.w_tax;
   str->D_TAX = (REAL) ora_str.newout.d_tax;
   strncpy (str->O_ENTRY_D, ora_str.newout.o_entry_d, 20);
   for (i = 0; i < ora_str.newout.o_ol_cnt; i++) {
      strncpy (str->item[i].I_NAME, ora_str.newout.i_name[i], 25);
      str->item[i].S_QUANTITY = ora_str.newout.s_quantity[i];
      str->item[i].brand_generic = ora_str.newout.brand_generic[i];
      str->item[i].I_PRICE = ora_str.newout.i_price[i]*100.0;  /* needs to be in
cents */
   }
   str->status = ((ora_str.newout.status[0] != '\0') ? E_INVALID_ITEM : OK);
}

/*****************************************************************************
 * Payment Query
 *****************************************************************************/

void
payment_transaction(str)
payment_trans *str;
{
   int i;

   struct paystruct ora_str;

   ora_str.payin.w_id = str->W_ID;
   ora_str.payin.d_id = str->D_ID;
   ora_str.payin.c_w_id = str->C_W_ID;
   ora_str.payin.c_d_id = str->C_D_ID;
   ora_str.payin.h_amount = str->H_AMOUNT; /* Amount in cents */
   ora_str.payin.bylastname = str->byname;
   if (ora_str.payin.bylastname) {
      ora_str.payin.c_id = 0;
      strncpy (ora_str.payin.c_last, str->C_LAST, 17);
      ora_str.payin.c_last[16] = '\0';
      for (i = 15; ((i >= 0) && (ora_str.payin.c_last[i] == ' ')); i--)
         ora_str.payin.c_last[i] = '\0';
   }
   else {
      ora_str.payin.c_id = str->C_ID;
      strcpy (ora_str.payin.c_last, " ");
   }
   retries = 0;
```

```
   numtrans++;
   if (TPCpay (&ora_str)) {
      str->status = E_DB_ERROR;
      return;
   } else {
      str->status = OK;
   }

   strncpy (str->W_STREET_1, ora_str.payout.w_street_1, 21);
   strncpy (str->W_STREET_2, ora_str.payout.w_street_2, 21);
   strncpy (str->W_CITY, ora_str.payout.w_city, 21);
   strncpy (str->W_STATE, ora_str.payout.w_state, 3);
   strncpy (str->W_ZIP, ora_str.payout.w_zip, 10);
   strncpy (str->D_STREET_1, ora_str.payout.d_street_1, 21);
   strncpy (str->D_STREET_2, ora_str.payout.d_street_2, 21);
   strncpy (str->D_CITY, ora_str.payout.d_city, 21);
   strncpy (str->D_STATE, ora_str.payout.d_state, 3);
   strncpy (str->D_ZIP, ora_str.payout.d_zip, 10);
   str->C_ID = ora_str.payout.c_id;
   strncpy (str->C_FIRST, ora_str.payout.c_first, 17);
   strncpy (str->C_MIDDLE, ora_str.payout.c_middle, 3);
   strncpy (str->C_LAST, ora_str.payout.c_last, 17);
   strncpy (str->C_STREET_1, ora_str.payout.c_street_1, 21);
   strncpy (str->C_STREET_2, ora_str.payout.c_street_2, 21);
   strncpy (str->C_CITY, ora_str.payout.c_city, 21);
   strncpy (str->C_STATE, ora_str.payout.c_state, 3);
   strncpy (str->C_ZIP, ora_str.payout.c_zip, 10);
   strncpy (str->C_PHONE, ora_str.payout.c_phone, 17);
   strncpy (str->C_SINCE, ora_str.payout.c_since, 11);

   strncpy (str->C_CREDIT, ora_str.payout.c_credit, 3);
   str->C_CREDIT_LIM = (MONEY) ora_str.payout.c_credit_lim*100.0; /* needs to be in
cents */
   str->C_DISCOUNT = (REAL) ora_str.payout.c_discount;
   str->C_BALANCE = (REAL) ora_str.payout.c_balance*100.0;  /* needs to be in cents
*/
   strncpy (str->C_DATA, ora_str.payout.c_data, 200);
   str->C_DATA[200] = '\0';
   strncpy (str->H_DATE, ora_str.payout.h_date, 20);
}

void
ordstat_transaction(str)
ordstat_trans *str;
{
   int i;

   struct ordstruct ora_str;

   ora_str.ordin.w_id = str->W_ID;
   ora_str.ordin.d_id = str->D_ID;
   ora_str.ordin.bylastname = str->byname;
   if (ora_str.ordin.bylastname) {
      ora_str.ordin.c_id = 0;
      strncpy (ora_str.ordin.c_last, str->C_LAST, 17);
      ora_str.ordin.c_last[16] = '\0';
      for (i = 15; ((i >= 0) && (ora_str.ordin.c_last[i] == ' ')); i--)
         ora_str.ordin.c_last[i] = '\0';
   }
   else {
      ora_str.ordin.c_id = str->C_ID;
      strcpy (ora_str.ordin.c_last, " ");
   }
   retries = 0;

   numtrans++;
   if (TPCord (&ora_str)) {
      str->status = E_DB_ERROR;
      return;
   } else {
      str->status = OK;
   }

   str->C_ID = ora_str.ordout.c_id;
   strncpy (str->C_LAST, ora_str.ordout.c_last, 17);
   strncpy (str->C_FIRST, ora_str.ordout.c_first, 17);
   strncpy (str->C_MIDDLE, ora_str.ordout.c_middle, 3);
   str->C_BALANCE = (MONEY) ora_str.ordout.c_balance*100.0; /* needs to be in cents
*/
   str->O_ID = ora_str.ordout.o_id;
   strncpy (str->O_ENTRY_DATE, ora_str.ordout.o_entry_d, 20);
   str->O_CARRIER_ID = ora_str.ordout.o_carrier_id;
   str->ol_cnt = ora_str.ordout.o_ol_cnt;
   for (i = 0; i < ora_str.ordout.o_ol_cnt; i++) {
      str->item[i].OL_SUPPLY_W_ID = ora_str.ordout.ol_supply_w_id[i];
      str->item[i].OL_I_ID = ora_str.ordout.ol_i_id[i];
      str->item[i].OL_QUANTITY = ora_str.ordout.ol_quantity[i];
      str->item[i].OL_AMOUNT = (MONEY) ora_str.ordout.ol_amount[i]*100.0;  /* needs
to be in cents */
      strncpy (str->item[i].OL_DELIVERY_DATE, ora_str.ordout.ol_delivery_d[i], 11);
   }
}

/*****************************************************************************
 * Delivery Query
 *****************************************************************************/

void delivery_transaction(str)
delivery_trans *str;
{
   double tr_end;
   int i;

   struct delstruct ora_str;

   ora_str.delin.w_id         = str->W_ID;
   ora_str.delin.o_carrier_id = str->O_CARRIER_ID;
   retries      = 0;

   numtrans++;
   if (TPCdel (&ora_str)) {
      str->status = E_DB_ERROR;
      return;
   } else {
      str->status = OK;
   }

   for (i = 0; i < 10; i++) {
      if (del_o_id[i] <= 0) {
           str->order[i].status = E_NOT_ENOUGH_ORDERS;
      } else {
           str->order[i].status = OK;
           str->order[i].O_ID = del_o_id[i];
      }
   }
}

/*****************************************************************************
 * Stock Level Query
 *****************************************************************************/

void stocklev_transaction(str)
stocklev_trans *str;
{

   struct stostruct ora_str;
   ora_str.stoin.w_id = str->W_ID;
   ora_str.stoin.d_id = str->D_ID;
   ora_str.stoin.threshold = str->threshold;
   retries = 0;

   numtrans++;
   if (TPCsto (&ora_str)) {
      str->status = E_DB_ERROR;
      return;
   } else {
      str->status = OK;
   }
   str->low_stock = ora_str.stoout.low_stock;
}
```

client/oracle/tpccpl.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: tpccpl.c 7030100.2 96/04/02 17:51:34 plai Generic<base> $ Copyr (c)
1994 Oracle";
#endif /* RCSID */


/*============================================================================+
 |         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA                |
 |                      OPEN SYSTEMS PERFORMANCE GROUP                        |
 |                         All Rights Reserved                                |
 +============================================================================+
 | FILENAME
 |    tpccpl.c
 | DESCRIPTION
 |    TPC-C transactions in PL/SQL.
 +============================================================================*/

#include <stdio.h>
#include <time.h>
#include "ora_tpcc.h"

#define SQLTXT "alter session set isolation_level = serializable"
#define SQLTXTTRC "alter session set sql_trace = true"
#define SQLTXTTIM "alter session set timed_statistics = true"

FILE *lfp;
FILE *fopen ();
#ifdef ORA_NT
extern double dpbtimef();
#define gettime dpbtimef
#else
double gettime ();
#endif
int proc_no = 0;
int logon = 0;
int new_init = 0;
int pay_init = 0;
int ord_init = 0;
int del_init = 0;
int sto_init = 0;
int execstatus;
int errcode;

OCIEnv *tpcenv;
OCIServer *tpcsrv;
OCIError *errhp;
OCISvcCtx *tpcsvc;
OCISession *tpcusr;
OCIStmt *curi;


/* for stock-level transaction */

int w_id;
int d_id;
int c_id;
int threshold;
int low_stock;

/* for delivery transaction */

int del_o_id[10];
int retries;

/* for order-status transaction */

int bylastname;
char c_last[17];
char c_first[17];
char c_middle[3];
double c_balance;
int o_id;
text o_entry_d[20];
ub4  datelen;
int o_carrier_id;
int o_ol_cnt;

int ol_supply_w_id[15];
int ol_i_id[15];
int ol_quantity[15];
int ol_amount[15];
ub4  ol_del_len[15];
text ol_delivery_d[15][11];

/* for payment transaction */

int c_w_id;
int c_d_id;
int h_amount;
char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
char w_zip[10];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
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];
ub4  sincelen;
text c_since_d[11];
float c_discount;
char c_credit[3];
int c_credit_lim;
char c_data[201];
ub4 hlen;
text h_date[20];

/* for new order transaction */

int nol_i_id[15];
int nol_supply_w_id[15];
int nol_quantity[15];
int nol_quanti10[15];
int nol_quanti91[15];
int nol_ytdqty[15];
int nol_amount[15];
int o_all_local;
float w_tax;
float d_tax;
float total_amount;
char i_name[15][25];
int s_quantity[15];
char brand_gen[15];
int i_price[15];
char brand_generic[15][1];
int status;
int tracelevel = 0;

OCIDate cr_date;
OCIDate c_since;
OCIDate o_entry_d_base;
OCIDate ol_d_base[15];
dvoid *xmem;

/* NewOrder Binding stuff */


int ocierror(fname, lineno, errhp, status)
char *fname;
int lineno;
OCIError *errhp;
sword status;
{
  text errbuf[512];
  ub4 buflen;
  sb4 errcode;
  sb4 lstat;
  ub4 recno=2;
  int rval = RECOVERR;
```

```
  switch (status) {
  case OCI_SUCCESS:
    break;
  case OCI_SUCCESS_WITH_INFO:
    message("Module %s Line %d\n", fname, lineno);
    message("Error - OCI_SUCCESS_WITH_INFO\n");
    lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                         (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    message("Error - %s\n", errbuf);
    break;
  case OCI_NEED_DATA:
    message("Module %s Line %d\n", fname, lineno);
    message("Error - OCI_NEED_DATA\n");
    rval = IRRECERR;
    break;
  case OCI_NO_DATA:
    message("Module %s Line %d\n", fname, lineno);
    message("Error - OCI_NO_DATA\n");
    rval = IRRECERR;
    break;
  case OCI_ERROR:
    lstat = OCIErrorGet (errhp, (ub4) 1,
                         (text *) NULL, &errcode, errbuf,
                         (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    if (errcode == NOT_SERIALIZABLE) return (errcode);
    while (lstat != OCI_NO_DATA)
    {
      message("Module %s Line %d\n", fname, lineno);
      message("Error - code %d - %s\n", errcode, errbuf);
      lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                           (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
    }
    rval = errcode;
    break;
  case OCI_INVALID_HANDLE:
    message("Module %s Line %d\n", fname, lineno);
    message("Error - OCI_INVALID_HANDLE\n");
    rval = IRRECERR;
    break;
  case OCI_STILL_EXECUTING:
    message("Module %s Line %d\n", fname, lineno);
    message("Error - OCI_STILL_EXECUTE\n");
    rval = IRRECERR;
    break;
  case OCI_CONTINUE:
    message("Module %s Line %d\n", fname, lineno);
    message("Error - OCI_CONTINUE\n");
    rval = IRRECERR;
    break;
  default:
    message("Module %s Line %d\n", fname, lineno);
    message("Status - %d\n", status);
    rval = IRRECERR;
    break;
  }

  if((rval != RECOVERR) &&
          (getenv("ABORT") != NULL))
 {
          extern int numtrans;

          message("ABORT ON ORACLE ERROR !!!! numtrans %d \n", numtrans);
          abort();
  }
  return (rval);
}


FILE  *vopen(fnam,mode)
char  *fnam;
char  *mode;
{
FILE *fd;

#ifdef  DEBUG
  message( "tkvuopen() fnam: %s,   mode: %s\n", fnam, mode);
#endif

    fd = fopen((char *)fnam,(char *)mode);
    if (!fd){
        message(" fopen on %s failed %d\n",fnam,fd);
        exit(-1);
    }
    return(fd);
}

int sqlfile(fnam,linebuf)
char   *fnam;
text   *linebuf;
{
FILE *fd;
int nulpt = 0;

#ifdef  DEBUG
  message( "sqlfile() fnam: %s,  linebuf: %#x\n", fnam, linebuf);
#endif

    fd = vopen(fnam,"r");
    while (fgets((char *)linebuf+nulpt, SQL_BUF_SIZE,fd))
    {
        nulpt = strlen((char *)linebuf);
    }
    return(nulpt);
}

#ifdef NOT
void vgetdate (unsigned char *oradt)
{
 struct tm *loctime;
 time_t  int_time;

 struct ORADATE {
 unsigned char      century;
 unsigned char      year;
 unsigned char      month;
 unsigned char      day;
 unsigned char      hour;
 unsigned char      minute;
 unsigned char      second;
 } Date;
int century;
int cnvrtOK;

/* assume convert is successful */
cnvrtOK = 1;

/* get the current date and time as an integer */
time( &int_time);

/* Convert the current date and time into local time */
loctime = localtime( &int_time);

century = (1900+loctime->tm_year) / 100;

Date.century = (unsigned char)(century + 100);
if (Date.century < 119 || Date.century > 120) cnvrtOK = 0;
Date.year  = (unsigned char)(loctime->tm_year+100);
if (Date.year < 100 || Date.year > 199) cnvrtOK = 0;
Date.month = (unsigned char)(loctime->tm_mon + 1);
if (Date.month < 1 || Date.month > 12) cnvrtOK = 0;
Date.day   = (unsigned char)loctime->tm_mday;
if (Date.day < 1 || Date.day > 31) cnvrtOK = 0;
Date.hour  = (unsigned char)(loctime->tm_hour + 1);
if (Date.hour < 1 || Date.hour > 24) cnvrtOK = 0;
Date.minute= (unsigned char)(loctime->tm_min + 1);
if (Date.minute < 1 || Date.minute > 60) cnvrtOK = 0;
Date.second= (unsigned char)(loctime->tm_sec + 1);
if (Date.second < 1 || Date.second > 60) cnvrtOK = 0;


if (cnvrtOK)
  memcpy(oradt,&Date,7);
  else
    *oradt = '\0';

 return;
```

```
}
void cvtdmy (unsigned char *oradt, char *outdate)
{

        struct ORADATE {
                unsigned char   century;
                unsigned char   year;
                unsigned char   month;
                unsigned char   day;
                unsigned char   hour;
                unsigned char   minute;
                unsigned char   second;
        } Date;


        int day,month,year;

        memcpy(&Date,oradt,7);

        year = (Date.century-100)*100 + Date.year-100;
        month = Date.month;
        day = Date.day;
        sprintf(outdate,"%02d-%02d-%4d\0",day,month,year);

        return;
}

void cvtdmyhms (unsigned char *oradt, char *outdate)
{

        struct ORADATE {
                unsigned char   century;
                unsigned char   year;
                unsigned char   month;
                unsigned char   day;
                unsigned char   hour;
                unsigned char   minute;
                unsigned char   second;
        } Date;


        int day,month,year;
        int hour,min,sec;

        memcpy(&Date,oradt,7);

        year = (Date.century-100)*100 + Date.year-100;
        month = Date.month;
        day = Date.day;
        hour = Date.hour - 1;
        min = Date.minute - 1;
        sec = Date.second - 1;

        sprintf(outdate,"%02d-%02d-%4d %02d:%02d:%02d\0",
                        day,month,year,hour,min,sec);

        return;
}
#endif


TPCexit ()

{

   if (new_init) {
      tkvcndone();
      new_init = 0;
   }
   if (pay_init) {
      tkvcpdone();
      pay_init = 0;
   }
   if (ord_init) {
      tkvcodone();
      ord_init = 0;
   }
   if (del_init) {
      tkvcddone();
      del_init = 0;
   }
   if (sto_init) {
      tkvcsdone();
      sto_init = 0;
   }


   OCIHandleFree((dvoid *)tpcusr, OCI_HTYPE_SESSION);
   OCIHandleFree((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX);
   OCIHandleFree((dvoid *)errhp, OCI_HTYPE_ERROR);
   OCIHandleFree((dvoid *)tpcsrv, OCI_HTYPE_SERVER);
   OCIHandleFree((dvoid *)tpcenv, OCI_HTYPE_ENV);

   if (lfp) {
      fclose (lfp);
      lfp = NULL;
   }

}



TPCinit (id, uid, pwd)

int id;
char *uid;
char *pwd;

{

   int i;
   char filename[40];
   text stmbuf[100];

   proc_no = id;

   OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoid *)0,0,0,0);
   OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid **)0);
   OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcsrv, OCI_HTYPE_SERVER, 0 , (dvoid
**)0);
   OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&errhp, OCI_HTYPE_ERROR, 0 , (dvoid
**)0);
   OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcsvc, OCI_HTYPE_SVCCTX, 0 , (dvoid
**)0);
   OCIERROR(errhp, OCIServerAttach(tpcsrv, errhp, (text *)0,0,OCI_DEFAULT));
   OCIAttrSet((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
(ub4)0,OCI_ATTR_SERVER, errhp);
   OCIHandleAlloc((dvoid *)tpcenv, (dvoid **)&tpcusr, OCI_HTYPE_SESSION, 0 , (dvoid
**)0);
   OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)uid,
(ub4)strlen(uid),OCI_ATTR_USERNAME, errhp);
   OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)pwd, (ub4)strlen(pwd),
                OCI_ATTR_PASSWORD, errhp);
   OCIERROR(errhp, OCISessionBegin(tpcsvc, errhp, tpcusr, OCI_CRED_RDBMS,
OCI_DEFAULT));

   OCIAttrSet(tpcsvc, OCI_HTYPE_SVCCTX, tpcusr, 0, OCI_ATTR_SESSION, errhp);

   {
          time_t pt;
          time(&pt);
          message("proc %d connected at %s", proc_no, ctime(&pt));
   }

   /* run all transaction in serializable mode */

   OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT);
   OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf), OCI_NTV_SYNTAX,
OCI_DEFAULT);
   OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
   OCIHandleFree(curi, OCI_HTYPE_STMT);


   if (tracelevel == 3) {
     OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
     memset(stmbuf,0,100);
     sprintf ((char *) stmbuf, SQLTXTTIM);
```

```
  OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                          OCI_NTV_SYNTAX, OCI_DEFAULT);
  OCIERROR(errhp, OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
  OCIHandleFree((dvoid *)curi, OCI_HTYPE_STMT);
}

logon = 1;

OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

if (tkvcninit ()) {  /* new order */
   TPCexit ();
   return (-1);
}
else
   new_init = 1;

if (tkvcpinit ()) {   /* payment */
   TPCexit ();
   return (-1);
}
else
   pay_init = 1;

if (tkvcoinit ()) {   /* order status */
   TPCexit ();
   return (-1);
}
else
   ord_init = 1;


if (tkvcdinit ()) {   /* delivery */
   TPCexit ();
   return (-1);
}
else
   del_init = 1;

if (tkvcsinit ()) {   /* stock level */
   TPCexit ();
   return (-1);
}
else
   sto_init = 1;

return (0);

}



TPCnew (str)

struct newstruct *str;

{

   int i;

   w_id = str->newin.w_id;
   d_id = str->newin.d_id;
   c_id = str->newin.c_id;
   for (i = 0; i < 15; i++) {
      nol_i_id[i] = str->newin.ol_i_id[i];
      nol_supply_w_id[i] = str->newin.ol_supply_w_id[i];
      nol_quantity[i] = str->newin.ol_quantity[i];
   }
   retries = 0;

   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

   if (str->newout.terror = tkvcn ()) {
      if (str->newout.terror != RECOVERR)
         str->newout.terror = IRRECERR;
      return (-1);
   }

   /* fill in date for o_entry_d from time in beginning of txn*/
    datelen = sizeof(o_entry_d);
    OCIERROR(errhp,
        OCIDateToText(errhp,&cr_date,(text*)FULLDATE,SIZ(FULLDATE),(text*)0,0,
                      &datelen,o_entry_d));

   str->newout.terror = NOERR;
   str->newout.o_id = o_id;
   str->newout.o_ol_cnt = o_ol_cnt;
   strncpy (str->newout.c_last, c_last, 17);
   strncpy (str->newout.c_credit, c_credit, 3);
   str->newout.c_discount = (float)(c_discount);
   str->newout.w_tax = (float)(w_tax);
   str->newout.d_tax = (float)(d_tax);
   strncpy (str->newout.o_entry_d, (char*)o_entry_d, 20);
   str->newout.total_amount = total_amount;
   for (i = 0; i < o_ol_cnt; i++) {
      strncpy (str->newout.i_name[i], i_name[i], 25);
      str->newout.s_quantity[i] = s_quantity[i];
      str->newout.brand_generic[i] = brand_generic[i][0];
      str->newout.i_price[i] = (float)(i_price[i])/100;
      str->newout.ol_amount[i] = (float)(nol_amount[i])/100;
   }
   if (status)
      strcpy (str->newout.status, "Item number is not valid");
   else
      str->newout.status[0] = '\0';
   str->newout.retry = retries;
   return (0);

}



TPCpay (str)

struct paystruct *str;

{

   w_id = str->payin.w_id;
   d_id = str->payin.d_id;
   c_w_id = str->payin.c_w_id;
   c_d_id = str->payin.c_d_id;
   h_amount = str->payin.h_amount;
   bylastname = str->payin.bylastname;

   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

   if (bylastname) {
      c_id = 0;
      strncpy (c_last, str->payin.c_last, 17);
   }
   else {
      c_id = str->payin.c_id;
      strcpy (c_last, " ");
   }
   retries = 0;

   if (str->payout.terror = tkvcp ()) {
      if (str->payout.terror != RECOVERR)
         str->payout.terror = IRRECERR;
      return (-1);
   }

        hlen=SIZ(h_date);
        OCIERROR(errhp,OCIDateToText(errhp,&cr_date,
                (text*)FULLDATE,strlen(FULLDATE),(text*)0,0,&hlen,h_date));

        sincelen=SIZ(c_since_d);
        OCIERROR(errhp,OCIDateToText(errhp,&c_since,
           (text*)SHORTDATE,strlen(SHORTDATE),(text*)0,0,&sincelen,c_since_d));



   str->payout.terror = NOERR;
   strncpy (str->payout.w_street_1, w_street_1, 21);
   strncpy (str->payout.w_street_2, w_street_2, 21);
   strncpy (str->payout.w_city, w_city, 21);
   strncpy (str->payout.w_state, w_state, 3);
```

```
   strncpy (str->payout.w_zip, w_zip, 10);
   strncpy (str->payout.d_street_1, d_street_1, 21);
   strncpy (str->payout.d_street_2, d_street_2, 21);
   strncpy (str->payout.d_city, d_city, 21);
   strncpy (str->payout.d_state, d_state, 3);
   strncpy (str->payout.d_zip, d_zip, 10);
   str->payout.c_id = c_id;
   strncpy (str->payout.c_first, c_first, 17);
   strncpy (str->payout.c_middle, c_middle, 3);
   strncpy (str->payout.c_last, c_last, 17);
   strncpy (str->payout.c_street_1, c_street_1, 21);
   strncpy (str->payout.c_street_2, c_street_2, 21);
   strncpy (str->payout.c_city, c_city, 21);
   strncpy (str->payout.c_state, c_state, 3);
   strncpy (str->payout.c_zip, c_zip, 10);
   strncpy (str->payout.c_phone, c_phone, 17);
   strncpy (str->payout.c_since, (char*)c_since_d, 11);
   strncpy (str->payout.c_credit, c_credit, 3);
   str->payout.c_credit_lim = (float)(c_credit_lim)/100;
   str->payout.c_discount = (float)(c_discount);
   str->payout.c_balance = (float)(c_balance)/100;
   strncpy (str->payout.c_data, c_data, 201);
   strncpy (str->payout.h_date, (char*)h_date, 20);
   str->payout.retry = retries;
   return (0);

}



TPCord (str)

struct ordstruct *str;

{

   int i;
   w_id = str->ordin.w_id;
   d_id = str->ordin.d_id;
   bylastname = str->ordin.bylastname;
   if (bylastname) {
      c_id = 0;
      strncpy (c_last, str->ordin.c_last, 17);
   }
   else {
      c_id = str->ordin.c_id;
      strcpy (c_last, " ");
   }
   retries = 0;

   if (str->ordout.terror = tkvco ()) {
      if (str->ordout.terror != RECOVERR)
         str->ordout.terror = IRRECERR;
      return (-1);
   }

   datelen = sizeof(o_entry_d);
   OCIERROR(errhp,

OCIDateToText(errhp,&o_entry_d_base,(text*)FULLDATE,SIZ(FULLDATE),(text*)0,0,
                     &datelen,o_entry_d));

   str->ordout.terror = NOERR;
   str->ordout.c_id = c_id;
   strncpy (str->ordout.c_last, c_last, 17);
   strncpy (str->ordout.c_first, c_first, 17);
   strncpy (str->ordout.c_middle, c_middle, 3);
   str->ordout.c_balance = c_balance/100;
   str->ordout.o_id = o_id;
   strncpy (str->ordout.o_entry_d, (char*)o_entry_d, 20);
   if ( o_carrier_id == 11 )
       str->ordout.o_carrier_id = 0;
   else
       str->ordout.o_carrier_id = o_carrier_id;
   str->ordout.o_ol_cnt = o_ol_cnt;
   for (i = 0; i < o_ol_cnt; i++) {
      ol_delivery_d[i][10] = '\0';
      if ( !strcmp((char*)ol_delivery_d[i],ORA_NULL_DATE) )
            strncpy((char*)ol_delivery_d[i],"NOT DELIVR",10);
      str->ordout.ol_supply_w_id[i] = ol_supply_w_id[i];
      str->ordout.ol_i_id[i] = ol_i_id[i];
      str->ordout.ol_quantity[i] = ol_quantity[i];
      str->ordout.ol_amount[i] = (float)(ol_amount[i])/100;
      strncpy (str->ordout.ol_delivery_d[i], (char*)ol_delivery_d[i], 11);
   }
   str->ordout.retry = retries;
   return (0);

}



TPCdel (str)

struct delstruct *str;

{

   double tr_end;
   int i;

   w_id = str->delin.w_id;
   o_carrier_id = str->delin.o_carrier_id;
   retries = 0;
   OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

   if (str->delout.terror = tkvcd ()) {
     if(str->delout.terror == DEL_ERROR)
       return DEL_ERROR;
      if (str->delout.terror != RECOVERR)
         str->delout.terror = IRRECERR;
      return (-1);
   }

   str->delout.terror = NOERR;
   str->delout.retry = retries;
   return (0);

}



TPCsto (str)

struct stostruct *str;

{

   w_id = str->stoin.w_id;
   d_id = str->stoin.d_id;
   threshold = str->stoin.threshold;
   retries = 0;

   if (str->stoout.terror = tkvcs ()) {
      if (str->stoout.terror != RECOVERR)
         str->stoout.terror = IRRECERR;
      return (-1);
   }

   str->stoout.terror = NOERR;
   str->stoout.low_stock = low_stock;
   str->stoout.retry = retries;
   return (0);

}
```

client/oracle/plnew.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: tkvcnew.c 21-apr-98.18:32:59 rdecker Exp $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*=============================================================+
 |         Copyright (c) 1996 , 1997, 1998 Oracle Corp, Redwood Shores, CA       |
```

```
|                    OPEN SYSTEMS PERFORMANCE GROUP                  |
|                         All Rights Reserved                        |
+===================================================================+
| FILENAME
|    plnew.c
| DESCRIPTION
|    OCI version (using PL/SQL stored procedure) of
|    NEW ORDER transaction in TPC-C benchmark.
+===================================================================*/

#include "ora_tpcc.h"

#ifdef PLSQLNO
#define SQLTXT2 "BEGIN initnew.new_init(:idx1arr); END;"
#else
#define SQLTXT2 "UPDATE stock SET s_order_cnt = s_order_cnt + 1, \
   s_ytd = s_ytd + :ol_quantity, s_remote_cnt = s_remote_cnt + :s_remote, \
   s_quantity = :s_quantity \
   WHERE rowid = :s_rowid"

#define SQLTXT3 "\
SELECT 0,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :10 AND s_w_id = :30 AND s_i_id = i_id UNION ALL \
SELECT 1,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :11 AND s_w_id = :31 AND s_i_id = i_id UNION ALL \
SELECT 2,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :12 AND s_w_id = :32 AND s_i_id = i_id UNION ALL \
SELECT 3,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :13 AND s_w_id = :33 AND s_i_id = i_id UNION ALL \
SELECT 4,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :14 AND s_w_id = :34 AND s_i_id = i_id UNION ALL \
SELECT 5,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :15 AND s_w_id = :35 AND s_i_id = i_id UNION ALL \
SELECT 6,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :16 AND s_w_id = :36 AND s_i_id = i_id UNION ALL \
SELECT 7,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :17 AND s_w_id = :37 AND s_i_id = i_id UNION ALL \
SELECT 8,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :18 AND s_w_id = :38 AND s_i_id = i_id UNION ALL \
SELECT 9,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :19 AND s_w_id = :39 AND s_i_id = i_id UNION ALL \
SELECT 10,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :20 AND s_w_id = :40 AND s_i_id = i_id UNION ALL \
SELECT 11,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :21 AND s_w_id = :41 AND s_i_id = i_id UNION ALL \
SELECT 12,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :22 AND s_w_id = :42 AND s_i_id = i_id UNION ALL \
SELECT 13,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :23 AND s_w_id = :43 AND s_i_id = i_id UNION ALL \
SELECT 14,stock.rowid,i_price,i_name,i_data,s_dist_%02d,s_data,s_quantity \
FROM item,stock WHERE i_id = :24 AND s_w_id = :44 AND s_i_id = i_id"


#define SQLTXT4 "INSERT INTO order_line \
   (ol_o_id, ol_d_id, ol_w_id, oL_number, ol_delivery_d, ol_i_id, \
    ol_supply_w_id, ol_quantity,ol_amount,ol_dist_info)\
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number, :null_date, :ol_i_id, :ol_supply_w_id, :ol_quantity, \
   :ol_amount, :ol_dist_info)"
#endif  /* PLSQLNO */

#define NITEMS 15
#define ROWIDLEN 20
#define OCIROWLEN 20

sb4 no_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
              dvoid **bufpp, ub4 *alenp, ub1 *piecep,
              dvoid **indpp)
{
  *bufpp = (dvoid*)0;
  *alenp =0;
  *indpp = (dvoid*)0;
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
struct newctx {
  sb2 nol_i_id_ind[NITEMS];
  sb2 nol_supply_w_id_ind[NITEMS];
  sb2 nol_quantity_ind[NITEMS];
  sb2 nol_amount_ind[NITEMS];
  sb2 i_name_ind[NITEMS];
  sb2 s_quantity_ind[NITEMS];
  sb2 i_price_ind[NITEMS];
  sb2 ol_w_id_ind[NITEMS];
  sb2 ol_d_id_ind[NITEMS];
  sb2 ol_o_id_ind[NITEMS];
  sb2 ol_number_ind[NITEMS];
  sb2 cons_ind[NITEMS];
  sb2 s_rowid_ind[NITEMS];
  sb2 s_remote_ind[NITEMS];
  sb2 s_quant_ind[NITEMS];
  sb2 i_data_ind[NITEMS];
  sb2 s_data_ind[NITEMS];
  sb2 s_dist_info_ind[NITEMS];
  sb2 ol_dist_info_ind[NITEMS];
  sb2 null_date_ind[NITEMS];
#ifdef PLSQLNO
  sb2 s_bg_ind[NITEMS];
#endif

  ub2 nol_i_id_len[NITEMS];
  ub2 nol_supply_w_id_len[NITEMS];
  ub2 nol_quantity_len[NITEMS];
  ub2 nol_amount_len[NITEMS];
  ub2 s_quantity_len[NITEMS];
  ub2 i_name_len[NITEMS];
  ub2 i_price_len[NITEMS];
  ub2 i_data_len[NITEMS];
  ub2 s_dist_info_len[NITEMS];
  ub2 s_data_len[NITEMS];
  ub2 ol_w_id_len[NITEMS];
  ub2 ol_d_id_len[NITEMS];
  ub2 ol_o_id_len[NITEMS];
  ub2 ol_number_len[NITEMS];
  ub2 cons_len[NITEMS];
  ub2 s_rowid_len[NITEMS];
  ub2 s_remote_len[NITEMS];
  ub2 s_quant_len[NITEMS];
  ub2 ol_dist_info_len[NITEMS];
  ub2 null_date_len[NITEMS];
#ifdef PLSQLNO
  ub2 s_bg_len[NITEMS];
#endif

  ub2 nol_i_id_rcode[NITEMS];
  ub2 nol_supply_w_id_rcode[NITEMS];
  ub2 nol_quantity_rcode[NITEMS];
  ub2 nol_amount_rcode[NITEMS];
  ub2 i_name_rcode[NITEMS];
  ub2 s_quantity_rcode[NITEMS];
  ub2 i_price_rcode[NITEMS];
  ub2 ol_w_id_rcode[NITEMS];
  ub2 ol_d_id_rcode[NITEMS];
  ub2 ol_o_id_rcode[NITEMS];
  ub2 ol_number_rcode[NITEMS];
  ub2 cons_rcode[NITEMS];
  ub2 s_rowid_rcode[NITEMS];
  ub2 s_remote_rcode[NITEMS];
  ub2 s_quant_rcode[NITEMS];
  ub2 i_data_rcode[NITEMS];
  ub2 s_data_rcode[NITEMS];
  ub2 s_dist_info_rcode[NITEMS];
  ub2 ol_dist_info_rcode[NITEMS];
  ub2 null_date_rcode[NITEMS];
#ifdef PLSQLNO
  ub2 s_bg_rcode[NITEMS];
#endif

  int ol_w_id[NITEMS];
  int ol_d_id[NITEMS];
  int ol_o_id[NITEMS];
  int ol_number[NITEMS];
  int cons[NITEMS];

  OCIRowid  *s_rowid_ptr[NITEMS];

  int s_remote[NITEMS];
  char i_data[NITEMS][51];
```

```
  char s_data[NITEMS][51];
  char s_dist_info[NITEMS][25];
  OCIDate null_date[NITEMS]; /* base date for null date entry */
  OCIStmt *curn1;
#ifdef PLSQLNO
  OCIBind *ol_i_id_bp;
  OCIBind *ol_supply_w_id_bp;
  OCIBind *i_price_bp;
  OCIBind *i_name_bp;
  OCIBind *s_bg_bp;
  OCIBind *s_data_bp;
  OCIBind *i_data_bp;
  ub4 nol_i_count;
  ub4 nol_s_count;
  ub4 nol_q_count;
  ub4 nol_item_count;
  ub4 nol_name_count;
  ub4 nol_qty_count;
  ub4 nol_bg_count;
  ub4 nol_am_count;
  ub4 s_remote_count;
  ub4 s_data_count;
  ub4 i_data_count;
#endif
  OCIStmt *curn2;
  OCIStmt *curn3[10];
  OCIBind *ol_i_id_bp4;
  OCIBind *ol_supply_w_id_bp4;
  OCIBind *ol_quantity_bp;
  OCIBind *ol_quantity_bp4;
  OCIBind *s_remote_bp;
  OCIBind *s_quantity_bp;
  OCIStmt *curn4;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *c_id_bp;
  OCIBind *o_all_local_bp;
  OCIBind *o_all_cnt_bp;
  OCIBind *w_tax_bp;
  OCIBind *d_tax_bp;
  OCIBind *o_id_bp;
  OCIBind *c_discount_bp;
  OCIBind *c_credit_bp;
  OCIBind *c_last_bp;
  OCIBind *retries_bp;
  OCIBind *cr_date_bp;
  OCIBind *s_rowid_bp;
  OCIBind *id_bp[10][15];
  OCIBind *sd_bp[10][15];
  OCIDefine *Dcons[10];
  OCIDefine *Ds_rowid[10];
  OCIDefine *Di_price[10];
  OCIDefine *Di_data[10];
  OCIDefine *Ds_dist_info[10];
  OCIDefine *Ds_data[10];
  OCIDefine *Ds_quantity[10];
  OCIDefine *Di_name[10];
  OCIBind *ol_o_id_bp;
  OCIBind *ol_d_id_bp;
  OCIBind *ol_w_id_bp;
  OCIBind *ol_number_bp;
  OCIBind *ol_amount_bp;
  OCIBind *ol_dist_info_bp;
  OCIBind *null_date_bp;

  sb2 w_id_ind;
  ub2 w_id_len;
  ub2 w_id_rc;

  sb2 d_id_ind;
  ub2 d_id_len;
  ub2 d_id_rc;

  sb2 c_id_ind;
  ub2 c_id_len;
  ub2 c_id_rc;

  sb2 o_all_local_ind;
  ub2 o_all_local_len;
  ub2 o_all_local_rc;

  sb2 o_ol_cnt_ind;
  ub2 o_ol_cnt_len;
  ub2 o_ol_cnt_rc;

  sb2 w_tax_ind;
  ub2 w_tax_len;
  ub2 w_tax_rc;

  sb2 d_tax_ind;
  ub2 d_tax_len;
  ub2 d_tax_rc;

  sb2 o_id_ind;
  ub2 o_id_len;
  ub2 o_id_rc;

  sb2 c_discount_ind;
  ub2 c_discount_len;
  ub2 c_discount_rc;

  sb2 c_credit_ind;
  ub2 c_credit_len;
  ub2 c_credit_rc;

  sb2 c_last_ind;
  ub2 c_last_len;
  ub2 c_last_rc;

  sb2 retries_ind;
  ub2 retries_len;
  ub2 retries_rc;

  sb2 cr_date_ind;
  ub2 cr_date_len;
  ub2 cr_date_rc;

  int cs;
  int norow;

/* context holders */
  int i_name_ctx;
  int i_data_ctx;
  int i_price_ctx;
  int s_data_ctx;
  int s_dist_info_ctx;
  int s_quantity_ctx;
};

typedef struct newctx newctx;

newctx *nctx;

tkvcninit ()
{

  int i, j;
  text stmbuf[16*1024];
  char id[4];
  char sd[4];

  nctx = (newctx *) malloc (sizeof(newctx));
  memset(nctx,(char)0,sizeof(newctx));
  nctx->cs = 1;
  nctx->norow=0;
  for(i=0;i<NITEMS;i++) {
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&nctx->s_rowid_ptr[i],
       OCI_DTYPE_ROWID,0,(dvoid**)0));
  }
  nctx->w_id_ind = TRUE;
  nctx->w_id_len = sizeof(w_id);
  nctx->d_id_ind = TRUE;
  nctx->d_id_len = sizeof(d_id);
  nctx->c_id_ind = TRUE;
  nctx->c_id_len = sizeof(c_id);
  nctx->o_all_local_ind = TRUE;
  nctx->o_all_local_len = sizeof(o_all_local);
  nctx->o_ol_cnt_ind = TRUE;
```

```
  nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
  nctx->w_tax_ind = TRUE;
  nctx->w_tax_len = 0;
  nctx->d_tax_ind = TRUE;
  nctx->d_tax_len = 0;
  nctx->o_id_ind = TRUE;
  nctx->o_id_len = sizeof(o_id);
  nctx->c_discount_ind = TRUE;
  nctx->c_discount_len = 0;
  nctx->c_credit_ind = TRUE;
  nctx->c_credit_len = 0;
  nctx->c_last_ind = TRUE;
  nctx->c_last_len = 0;
  nctx->retries_ind = TRUE;
  nctx->retries_len = sizeof(retries);
  nctx->cr_date_ind = TRUE;
  nctx->cr_date_len = sizeof(cr_date);

  /* open first cursor */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&nctx->curn1),
           OCI_HTYPE_STMT, 0, (dvoid**)0));
#ifdef PLSQLNO
  sqlfile(ORA_BLOCK_PATH "tkvcpnew.sql",stmbuf);
#else
  sqlfile(ORA_BLOCK_PATH "tkvcbnew.sql",stmbuf);
#endif
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn1, errhp, stmbuf, strlen((char *)stmbuf),
                                OCI_NTV_SYNTAX, OCI_DEFAULT));

  /* bind variables */

  OCIBNDR(nctx->curn1, nctx->w_id_bp, errhp, ":w_id",ADR(w_id),SIZ(w_id),
          SQLT_INT, &nctx->w_id_ind, &nctx->w_id_len, &nctx->w_id_rc);
  OCIBNDR(nctx->curn1, nctx->d_id_bp, errhp, ":d_id",ADR(d_id),SIZ(d_id),
          SQLT_INT, &nctx->d_id_ind, &nctx->d_id_len, &nctx->d_id_rc);
  OCIBNDR(nctx->curn1, nctx->c_id_bp, errhp, ":c_id",ADR(c_id),SIZ(c_id),
          SQLT_INT, &nctx->c_id_ind, &nctx->c_id_len, &nctx->c_id_rc);
  OCIBNDR(nctx->curn1, nctx->o_all_local_bp, errhp, ":o_all_local",
          ADR(o_all_local), SIZ(o_all_local),SQLT_INT, &nctx->o_all_local_ind,
          &nctx->o_all_local_len, &nctx->o_all_local_rc);
  OCIBNDR(nctx->curn1, nctx->o_all_cnt_bp, errhp, ":o_ol_cnt",ADR(o_ol_cnt),
          SIZ(o_ol_cnt),SQLT_INT,&nctx->o_ol_cnt_ind, &nctx->o_ol_cnt_len,
          &nctx->o_ol_cnt_rc);
  OCIBNDR(nctx->curn1, nctx->w_tax_bp, errhp, ":w_tax",ADR(w_tax),SIZ(w_tax),
          SQLT_FLT, &nctx->w_tax_ind, &nctx->w_tax_len, &nctx->w_tax_rc);
  OCIBNDR(nctx->curn1, nctx->d_tax_bp, errhp, ":d_tax",ADR(d_tax),SIZ(d_tax),
          SQLT_FLT, &nctx->d_tax_ind, &nctx->d_tax_len, &nctx->d_tax_rc);
  OCIBNDR(nctx->curn1, nctx->o_id_bp, errhp, ":o_id",ADR(o_id),SIZ(o_id),
          SQLT_INT, &nctx->o_id_ind, &nctx->o_id_len, &nctx->o_id_rc);
  OCIBNDR(nctx->curn1, nctx->c_discount_bp, errhp, ":c_discount",
          ADR(c_discount), SIZ(c_discount),SQLT_FLT,
      &nctx->c_discount_ind, &nctx->c_discount_len, &nctx->c_discount_rc);
  OCIBNDR(nctx->curn1, nctx->c_credit_bp, errhp, ":c_credit",c_credit,
          SIZ(c_credit),SQLT_CHR,
      &nctx->c_credit_ind, &nctx->c_credit_len, &nctx->c_credit_rc);
  OCIBNDR(nctx->curn1, nctx->c_last_bp, errhp, ":c_last",c_last,SIZ(c_last),
          SQLT_STR, &nctx->c_last_ind, &nctx->c_last_len, &nctx->c_last_rc);
  OCIBNDR(nctx->curn1, nctx->retries_bp, errhp, ":retry",ADR(retries),
          SIZ(retries),SQLT_INT,
      &nctx->retries_ind, &nctx->retries_len, &nctx->retries_rc);
  OCIBNDR(nctx->curn1, nctx->cr_date_bp, errhp, ":cr_date",&cr_date,SIZ(OCIDate),
 SQLT_ODT, &nctx->cr_date_ind, &nctx->cr_date_len, &nctx->cr_date_rc);

#ifdef PLSQLNO
  OCIBNDRAA(nctx->curn1, nctx->ol_i_id_bp,errhp,":ol_i_id",nol_i_id,
           SIZ(int), SQLT_INT, nctx->nol_i_id_ind,nctx->nol_i_id_len,
           nctx->nol_i_id_rcode,NITEMS,&nctx->nol_i_count);
  OCIBNDRAA(nctx->curn1, nctx->ol_supply_w_id_bp, errhp, ":ol_supply_w_id",
           nol_supply_w_id,SIZ(int),SQLT_INT, nctx->nol_supply_w_id_ind,
           nctx->nol_supply_w_id_len, nctx->nol_supply_w_id_rcode,
           NITEMS, &nctx->nol_s_count);
  OCIBNDRAA(nctx->curn1, nctx->ol_quantity_bp,errhp,":ol_quantity",nol_quantity,
           SIZ(int),SQLT_INT,nctx->nol_quantity_ind,nctx->nol_quantity_len,
           nctx->nol_quantity_rcode,NITEMS,&nctx->nol_q_count);
  OCIBNDRAA(nctx->curn1, nctx->i_price_bp,errhp,":i_price",i_price,SIZ(int),
            SQLT_INT, nctx->i_price_ind,nctx->i_price_len,nctx->i_price_rcode,
            NITEMS, &nctx->nol_item_count);
  OCIBNDRAA(nctx->curn1, nctx->i_name_bp,errhp,":i_name",i_name,
             SIZ(i_name[0]),SQLT_STR, nctx->i_name_ind,nctx->i_name_len,
             nctx->i_name_rcode,NITEMS,&nctx->nol_name_count);
  OCIBNDRAA(nctx->curn1, nctx->s_quantity_bp,errhp,":s_quantity",s_quantity,
           SIZ(int), SQLT_INT,nctx->s_quant_ind,nctx->s_quant_len,
           nctx->s_quant_rcode,NITEMS,&nctx->nol_qty_count);
  OCIBNDRAA(nctx->curn1, nctx->s_bg_bp,errhp,":brand_generic",brand_generic,
           SIZ(char), SQLT_CHR,nctx->s_bg_ind,nctx->s_bg_len,
           nctx->s_bg_rcode,NITEMS,&nctx->nol_bg_count);
  OCIBNDRAA(nctx->curn1, nctx->ol_amount_bp,errhp,":ol_amount",nol_amount,
           SIZ(int),SQLT_INT, nctx->nol_amount_ind,nctx->nol_amount_len,
           nctx->nol_amount_rcode,NITEMS,&nctx->nol_am_count);
  OCIBNDRAA(nctx->curn1, nctx->s_remote_bp,errhp,":s_remote",nctx->s_remote,
           SIZ(int),SQLT_INT, nctx->s_remote_ind,nctx->s_remote_len,
           nctx->s_remote_rcode,NITEMS,&nctx->s_remote_count);

  /* open second cursor */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn2), OCI_HTYPE_STMT,
                                        0, (dvoid**)0));
  sprintf ((char *) stmbuf, SQLTXT2);
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn2, errhp, stmbuf,
           strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));


  /* execute second cursor to init newinit package */
  {
    int idx1arr[NITEMS];
    OCIBind *idx1arr_bp;
    ub2 idx1arr_len[NITEMS];
    ub2 idx1arr_rcode[NITEMS];
    sb2 idx1arr_ind[NITEMS];
    ub4 idx1arr_count;
    ub2 idx;

    for (idx = 0; idx < NITEMS; idx++) {
      idx1arr[idx] = idx + 1;
      idx1arr_ind[idx] = TRUE;
      idx1arr_len[idx] = sizeof(int);
    }
    idx1arr_count = NITEMS;
          o_ol_cnt = NITEMS;


    /* Bind array */
    OCIBNDRAA(nctx->curn2, idx1arr_bp,errhp,":idx1arr",idx1arr,
             SIZ(int), SQLT_INT, idx1arr_ind,idx1arr_len,
             idx1arr_rcode,NITEMS,&idx1arr_count);

     execstatus = OCIStmtExecute(tpcsvc,nctx->curn2,errhp,1,0,0,0,OCI_DEFAULT);
     if(execstatus != OCI_SUCCESS) {
       OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
       errcode = OCIERROR(errhp,execstatus);
       return -1;
     }
   }
#else

  /* open second cursor */
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn2), OCI_HTYPE_STMT,
                                        0, (dvoid**)0));
  sprintf ((char *) stmbuf, SQLTXT2);
  OCIERROR(errhp,OCIStmtPrepare(nctx->curn2, errhp, stmbuf,
           strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));


  /* bind variables */

  OCIBNDRA(nctx->curn2, nctx->s_quantity_bp,errhp,":s_quantity",s_quantity,
           SIZ(int), SQLT_INT,nctx->s_quant_ind,nctx->s_quant_len,
           nctx->s_quant_rcode);
  OCIBNDRA(nctx->curn2, nctx->s_rowid_bp, errhp, ":s_rowid",nctx->s_rowid_ptr,
           sizeof(nctx->s_rowid_ptr[0]),SQLT_RDD,nctx->s_rowid_ind,
             nctx->s_rowid_len,nctx->s_rowid_rcode);
  OCIBNDRA(nctx->curn2, nctx->ol_quantity_bp,errhp,":ol_quantity",nol_quantity,
             SIZ(int),SQLT_INT,nctx->nol_quantity_ind,nctx->nol_quantity_len,
             nctx->nol_quantity_rcode);
  OCIBNDRA(nctx->curn2, nctx->s_remote_bp, errhp, ":s_remote",nctx->s_remote,
           SIZ(int), SQLT_INT,nctx->s_remote_ind,nctx->s_remote_len,
             nctx->s_remote_rcode);
```

```
/* open third cursor and bind variables */

for (i = 0; i < 10; i++)
{
  j = i + 1;
  OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)&((nctx->curn3)[i]),
                                        OCI_HTYPE_STMT, 0, (dvoid**)0));

  sprintf ((char *) stmbuf, SQLTXT3, j, j, j, j, j, j, j, j, j, j, j, j,
           j, j, j);

  OCIERROR(errhp,OCIStmtPrepare((nctx->curn3)[i], errhp, stmbuf,
                                strlen((char *)stmbuf),OCI_NTV_SYNTAX,
                                OCI_DEFAULT));
  OCIERROR(errhp,
        OCIAttrSet(nctx->curn3[i],OCI_HTYPE_STMT,(dvoid*)&nctx->norow,0,
              OCI_ATTR_PREFETCH_ROWS,errhp));
  for (j = 0; j < NITEMS; j++)
  {
    sprintf (id, ":%d", j + 10);
    sprintf (sd, ":%d", j + 30);
    OCIBNDRA((nctx->curn3)[i],(nctx->id_bp)[i][j],errhp,id,ADR(nol_i_id[j]),
              SIZ(int),SQLT_INT,
              &nctx->nol_i_id_ind[j],&nctx->nol_i_id_len[j],
              &nctx->nol_i_id_rcode[j]);
    OCIBNDRA((nctx->curn3)[i],(nctx->sd_bp)[i][j],errhp,sd,
              ADR(nol_supply_w_id[j]),SIZ(int),SQLT_INT,
              &nctx->nol_supply_w_id_ind[j],&nctx->nol_supply_w_id_len[j],
              &nctx->nol_supply_w_id_rcode[j]);
    nctx->nol_i_id_ind[j] = NA;
    nctx->nol_supply_w_id_ind[j] = NA;
    nctx->nol_i_id_len[j] = sizeof(int);
    nctx->nol_supply_w_id_len[j] = sizeof(int);
  }

  OCIDEF((nctx->curn3)[i],(nctx->Dcons)[i],errhp,1,&(nctx->cons[0]),
           SIZ(nctx->cons[0]),SQLT_INT);
  OCIDEF((nctx->curn3)[i], (nctx->Ds_rowid)[i],errhp,2,
          nctx->s_rowid_ptr, sizeof(nctx->s_rowid_ptr[0]), SQLT_RDD);
  OCIDEF((nctx->curn3)[i], (nctx->Di_price)[i],errhp,3,i_price,SIZ(int),
          SQLT_INT);

  OCIDFNRA((nctx->curn3)[i], (nctx->Di_name)[i],errhp,4,i_name,
          SIZ(i_name[0]),SQLT_STR, nctx->i_name_ind,nctx->i_name_len,
          nctx->i_name_rcode);
  OCIDFNRA((nctx->curn3)[i], (nctx->Di_data)[i],errhp,5,nctx->i_data,
            SIZ(nctx->i_data[0]), SQLT_STR, NULL,nctx->i_data_len,NULL);
  OCIDFNRA((nctx->curn3)[i], (nctx->Ds_dist_info)[i],errhp,6,
          nctx->s_dist_info, SIZ(nctx->s_dist_info[0]),SQLT_STR,
          NULL,nctx->s_dist_info_len, NULL);
  OCIDFNRA((nctx->curn3)[i],(nctx->Ds_data)[i],errhp,7,nctx->s_data,
            SIZ(nctx->s_data[0]),SQLT_STR,NULL,nctx->s_data_len,NULL);
  OCIDEF((nctx->curn3)[i],(nctx->Ds_quantity)[i],errhp,8,s_quantity,
            SIZ(int),SQLT_INT);

 }

/* open fourth cursor */
OCIHandleAlloc(tpcenv, (dvoid **)(&nctx->curn4), OCI_HTYPE_STMT, 0,
               (dvoid**)0);
sprintf ((char *) stmbuf, SQLTXT4);
OCIStmtPrepare(nctx->curn4, errhp, stmbuf, strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX, OCI_DEFAULT);
/* bind variables */

OCIBNDRA(nctx->curn4, nctx->ol_o_id_bp,errhp,":ol_o_id",nctx->ol_o_id,
         SIZ(int),SQLT_INT, NULL,nctx->ol_o_id_len,
         NULL);
OCIBNDRA(nctx->curn4, nctx->ol_d_id_bp,errhp,":ol_d_id",nctx->ol_d_id,
         SIZ(int),SQLT_INT, NULL,nctx->ol_d_id_len,
         NULL);
OCIBNDRA(nctx->curn4, nctx->ol_w_id_bp,errhp,":ol_w_id",nctx->ol_w_id,
         SIZ(int),SQLT_INT, NULL,nctx->ol_w_id_len,
         NULL);
OCIBNDRA(nctx->curn4, nctx->ol_number_bp,errhp,":ol_number",nctx->ol_number,
         SIZ(int),SQLT_INT, NULL,nctx->ol_number_len,
         NULL);
OCIBNDRA(nctx->curn4, nctx->ol_i_id_bp4,errhp,":ol_i_id",nol_i_id,SIZ(int),
         SQLT_INT, NULL,nctx->nol_i_id_len, NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_supply_w_id_bp4,errhp,":ol_supply_w_id",
           nol_supply_w_id,SIZ(int),SQLT_INT, NULL,
           nctx->nol_supply_w_id_len, NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_quantity_bp4,errhp,":ol_quantity",nol_quantity,
           SIZ(int),SQLT_INT, NULL,nctx->nol_quantity_len,
             NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_amount_bp,errhp,":ol_amount",nol_amount,
           SIZ(int),SQLT_INT, NULL,nctx->nol_amount_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->ol_dist_info_bp,errhp,":ol_dist_info",
           nctx->s_dist_info, SIZ(nctx->s_dist_info[0]),SQLT_AFC,
           NULL, nctx->ol_dist_info_len,
           NULL);
  OCIBNDRA(nctx->curn4, nctx->null_date_bp,errhp,":null_date",nctx->null_date,
           SIZ(OCIDate), SQLT_ODT,NULL,
           nctx->null_date_len, NULL);

  /* set up the null date Null date is 01-01-1811 */
  for (i=0;i<NITEMS;i++)
  {
     OCIDateSetDate(&nctx->null_date[i],(sb2)1811,(ub1)1,(ub1)1);
  }
#endif

  return (0);

}



tkvcn ()
{

   int i, j, k;
   int rpc, rpc3, rowoff, iters,rcount;
   ub4 flags;
   int failed = 0;

retry:

   status = 0;                        /* number of invalid items */

   /* get number of order lines, and check if all are local */

   o_ol_cnt = NITEMS;
   o_all_local = 1;
   for (i = 0; i < NITEMS; i++) {
      if (nol_i_id[i] == 0) {
         o_ol_cnt = i;
         break;
      }
      if (nol_supply_w_id[i] != w_id) {
         nctx->s_remote[i] = 1;
         o_all_local = 0;
      }
      else
         nctx->s_remote[i] = 0;
   }

   nctx->w_id_ind = TRUE;
   nctx->w_id_len = sizeof(w_id);
   nctx->d_id_ind = TRUE;
   nctx->d_id_len = sizeof(d_id);
   nctx->c_id_ind = TRUE;
   nctx->c_id_len = sizeof(c_id);
   nctx->o_all_local_ind = TRUE;
   nctx->o_all_local_len = sizeof(o_all_local);
   nctx->o_ol_cnt_ind = TRUE;
   nctx->o_ol_cnt_len = sizeof(o_ol_cnt);
   nctx->w_tax_ind = TRUE;
   nctx->w_tax_len = 0;
   nctx->d_tax_ind = TRUE;
   nctx->d_tax_len = 0;
   nctx->o_id_ind = TRUE;
   nctx->o_id_len = sizeof(o_id);
   nctx->c_discount_ind = TRUE;
   nctx->c_discount_len = 0;
   nctx->c_credit_ind = TRUE;
   nctx->c_credit_len = 0;
```

```
   nctx->c_last_ind = TRUE;
   nctx->c_last_len = 0;
   nctx->retries_ind = TRUE;
   nctx->retries_len = sizeof(retries);
   nctx->cr_date_ind = TRUE;
   nctx->cr_date_len = sizeof(cr_date);
#ifdef PLSQLNO
   /* this is the row count */
   rcount = o_ol_cnt;
   nctx->nol_i_count = o_ol_cnt;
   nctx->nol_q_count = o_ol_cnt;
   nctx->nol_s_count = o_ol_cnt;
   nctx->s_remote_count = o_ol_cnt;

   nctx->nol_qty_count  = 0;
   nctx->nol_bg_count = 0;
   nctx->nol_item_count = 0;
   nctx->nol_name_count = 0;
   nctx->nol_am_count   = 0;
   /* following not relevant */
   nctx->s_data_count = o_ol_cnt;
   nctx->i_data_count = o_ol_cnt;

   /* initialization for array operations */
   for (i = 0; i < o_ol_cnt; i++) {
      nctx->ol_w_id[i] = w_id;
      nctx->ol_d_id[i] = d_id;
      nctx->ol_number[i] = i + 1;
      nctx->null_date_ind[i]= TRUE;
      nctx->nol_i_id_ind[i] = 0;
      nctx->nol_supply_w_id_ind[i] = TRUE;
      nctx->nol_quantity_ind[i] = TRUE;
      nctx->nol_amount_ind[i] = TRUE;
      nctx->ol_w_id_ind[i] = TRUE;
      nctx->ol_d_id_ind[i] = TRUE;
      nctx->ol_o_id_ind[i] = TRUE;
      nctx->ol_number_ind[i] = TRUE;
      nctx->ol_dist_info_ind[i] = TRUE;
      nctx->s_remote_ind[i] = TRUE;
      nctx->s_data_ind[i] = TRUE;
      nctx->i_data_ind[i] = TRUE;
      nctx->s_quant_ind[i] = TRUE;
      nctx->s_bg_ind[i] = TRUE;
      nctx->cons_ind[i] = TRUE;
      nctx->s_rowid_ind[i] = TRUE;
      nctx->nol_i_id_len[i] = sizeof(int);
      nctx->nol_supply_w_id_len[i] = sizeof(int);
      nctx->nol_quantity_len[i] = sizeof(int);
      nctx->nol_amount_len[i] = sizeof(int);
      nctx->ol_w_id_len[i] = sizeof(int);
      nctx->ol_d_id_len[i] = sizeof(int);
      nctx->ol_o_id_len[i] = sizeof(int);
      nctx->ol_number_len[i] = sizeof(int);
      nctx->ol_dist_info_len[i] = nctx->s_dist_info_len[i];
      nctx->null_date_len[i]=sizeof(OCIDate);
      nctx->s_remote_len[i] = sizeof(int);
      nctx->s_data_len[i] = sizeof(int);
      nctx->i_data_len[i] = sizeof(int);
      nctx->s_quant_len[i] = sizeof(int);
      nctx->s_rowid_len[i] = sizeof(nctx->s_rowid_ptr[0]);
      nctx->cons_len[i] = sizeof(int);
      nctx->i_name_len[i]=0;
      nctx->s_bg_len[i] = 0;
   }
   for (i = o_ol_cnt; i < NITEMS; i++) {
      nctx->nol_i_id_ind[i] = NA;
      nctx->nol_supply_w_id_ind[i] = NA;
      nctx->nol_quantity_ind[i] = NA;
      nctx->nol_amount_ind[i] = NA;
      nctx->ol_w_id_ind[i] = NA;
      nctx->ol_d_id_ind[i] = NA;
      nctx->ol_o_id_ind[i] = NA;
      nctx->ol_number_ind[i] = NA;
      nctx->ol_dist_info_ind[i] = NA;
      nctx->null_date_ind[i]= NA;
      nctx->s_remote_ind[i] = NA;
      nctx->s_data_ind[i] = NA;
      nctx->i_data_ind[i] = NA;
      nctx->s_quant_ind[i] = NA;
      nctx->s_bg_ind[i] = NA;
      nctx->cons_ind[i] = NA;
      nctx->s_rowid_ind[i] = NA;

      nctx->nol_i_id_len[i] = 0;
      nctx->nol_supply_w_id_len[i] = 0;
      nctx->nol_quantity_len[i] = 0;
      nctx->nol_amount_len[i] = 0;
      nctx->ol_w_id_len[i] = 0;
      nctx->ol_d_id_len[i] = 0;
      nctx->ol_o_id_len[i] = 0;
      nctx->ol_number_len[i] = 0;
      nctx->ol_dist_info_len[i] = 0;
      nctx->null_date_len[i]= 0;
      nctx->s_remote_len[i] = 0;
      nctx->i_data_len[i] = 0;
      nctx->s_data_len[i] = 0;
      nctx->s_quant_len[i] = 0;
      nctx->s_rowid_len[i] = 0;
      nctx->cons_len[i] = 0;
      nctx->i_name_len[i]=0;
      nctx->s_bg_len[i] = 0;
   }

   execstatus = OCIStmtExecute(tpcsvc,nctx->curn1,errhp,1,0,0,0,
                                       OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);

#else
   execstatus = OCIStmtExecute(tpcsvc,nctx->curn1,errhp,1,0,0,0,OCI_DEFAULT);
#endif


   if(execstatus != OCI_SUCCESS) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
          retries++;
          goto retry;
     } else if (errcode == RECOVERR) {
          retries++;
          goto retry;
     }
     else {
          return -1;
     }
   }

#ifdef PLSQLNO
   /* did the txn succeed ? */
   if (rcount != o_ol_cnt)
   {
     status = rcount - o_ol_cnt;
     o_ol_cnt = rcount;
   }
#endif

#ifdef DEBUG
   printf("w_id = %d, d_id = %d, c_id = %d\n",w_id, d_id, c_id);
#endif


#ifndef PLSQLNO
   /* initialization for array operations */

   for (i = 0; i < o_ol_cnt; i++) {
      nctx->ol_w_id[i] = w_id;
      nctx->ol_d_id[i] = d_id;
      nctx->ol_number[i] = i + 1;
      nctx->null_date_ind[i]= TRUE;
      nctx->nol_i_id_ind[i] = TRUE;
      nctx->nol_supply_w_id_ind[i] = TRUE;
      nctx->nol_quantity_ind[i] = TRUE;
      nctx->nol_amount_ind[i] = TRUE;
      nctx->ol_w_id_ind[i] = TRUE;
      nctx->ol_d_id_ind[i] = TRUE;
      nctx->ol_o_id_ind[i] = TRUE;
      nctx->ol_number_ind[i] = TRUE;
      nctx->ol_dist_info_ind[i] = TRUE;
      nctx->s_remote_ind[i] = TRUE;
```

```
   nctx->s_quant_ind[i] = TRUE;
   nctx->cons_ind[i] = TRUE;
   nctx->s_rowid_ind[i] = TRUE;

   nctx->nol_i_id_len[i] = sizeof(int);
   nctx->nol_supply_w_id_len[i] = sizeof(int);
   nctx->nol_quantity_len[i] = sizeof(int);
   nctx->nol_amount_len[i] = sizeof(int);
   nctx->ol_w_id_len[i] = sizeof(int);
   nctx->ol_d_id_len[i] = sizeof(int);
   nctx->ol_o_id_len[i] = sizeof(int);
   nctx->ol_number_len[i] = sizeof(int);
   nctx->ol_dist_info_len[i] = nctx->s_dist_info_len[i];
   nctx->null_date_len[i]=sizeof(OCIDate);
   nctx->s_remote_len[i] = sizeof(int);
   nctx->s_quant_len[i] = sizeof(int);
   nctx->s_rowid_len[i] = sizeof(nctx->s_rowid_ptr[0]);
   nctx->cons_len[i] = sizeof(int);
}
for (i = o_ol_cnt; i < NITEMS; i++) {
   nctx->nol_i_id_ind[i] = NA;
   nctx->nol_supply_w_id_ind[i] = NA;
   nctx->nol_quantity_ind[i] = NA;
   nctx->nol_amount_ind[i] = NA;
   nctx->ol_w_id_ind[i] = NA;
   nctx->ol_d_id_ind[i] = NA;
   nctx->ol_o_id_ind[i] = NA;
   nctx->ol_number_ind[i] = NA;
   nctx->ol_dist_info_ind[i] = NA;
   nctx->null_date_ind[i]= NA;
   nctx->s_remote_ind[i] = NA;
   nctx->s_quant_ind[i] = NA;
   nctx->cons_ind[i] = NA;
   nctx->s_rowid_ind[i] = NA;

   nctx->nol_i_id_len[i] = 0;
   nctx->nol_supply_w_id_len[i] = 0;
   nctx->nol_quantity_len[i] = 0;
   nctx->nol_amount_len[i] = 0;
   nctx->ol_w_id_len[i] = 0;
   nctx->ol_d_id_len[i] = 0;
   nctx->ol_o_id_len[i] = 0;
   nctx->ol_number_len[i] = 0;
   nctx->ol_dist_info_len[i] = 0;
   nctx->null_date_len[i]= 0;
   nctx->s_remote_len[i] = 0;
   nctx->s_quant_len[i] = 0;
   nctx->s_rowid_len[i] = 0;
   nctx->cons_len[i] = 0;
}

rpc3 = SelItemStk ();
if (rpc3 == -2)
   goto retry;
else if (rpc3 == -1)
   return (-1);

/* compute order line amounts, total amount and stock quantities */

total_amount = 0.0;
for (i = 0; i < o_ol_cnt; i++)
  {
   nctx->ol_o_id[i] = o_id;
   if (nctx->nol_i_id_ind[i] != NA) {
      s_quantity[i] -= nol_quantity[i];
      if (s_quantity[i] < 10)
         s_quantity[i] += 91;
      nol_amount[i] = (nol_quantity[i] * i_price[i]);
      total_amount += nol_amount[i];
      if (strstr (nctx->i_data[i], "ORIGINAL") &&
          strstr (nctx->s_data[i], "ORIGINAL"))
         brand_gen[i] = 'B';
      else
         brand_gen[i] = 'G';
   }

 }
total_amount *= (1.0 - c_discount) * (1.0 + d_tax + w_tax);
total_amount = total_amount/100;

   rpc = UpdStk2 ();
   if (rpc == -2)
      goto retry;
   else if (rpc == -1)
      return (-1);

   /* error processing - will keep it separated for readablity */
   /* number of items selected != number of stock updated */

   if (rpc3 != rpc) {
      message ("Error in TPC-C server %d: %d rows of item read, ",
               proc_no, rpc3);
      message ("                          but %d rows of stock updated\n", rpc);
      /* rollback */
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (-1);
   }


/* common code for insert into order_line */
   for (i=0; i< o_ol_cnt; i++)  /* move district info in place */
   {
     nctx->ol_dist_info_len[i]=nctx->s_dist_info_len[i];
   }

   /* array insert into order line table */
   flags= (status ? OCI_DEFAULT : (OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS));
   if ((o_ol_cnt - status) > 0)
   {
      execstatus = OCIStmtExecute(tpcsvc,nctx->curn4,errhp,o_ol_cnt - status,
                                  0,0,0,flags);
      if(execstatus != OCI_SUCCESS) {
        OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
        errcode = OCIERROR(errhp,execstatus);
        if(errcode == NOT_SERIALIZABLE) {
           retries++;
           goto retry;
        } else if (errcode == RECOVERR) {
           retries++;
           goto retry;
        } else {
           return -1;
        }
      }
      OCIAttrGet(nctx->curn4,OCI_HTYPE_STMT,&rcount,NULL,
                    OCI_ATTR_ROW_COUNT, errhp);
      if (rcount != (o_ol_cnt - status))
      {
       message ("Error in TPC-C server %d: array insert failed\n",
                  proc_no);
       /* rollback */
       OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
       return (-1);
     }
   }

   /* commit if no invalid item */

   if (status) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     fflush(stdout);
   }

#endif
   return (0);

}



void tkvcndone ()

{

   int i;

   if (nctx)
   {
```

```
      OCIHandleFree((dvoid *)nctx->curn1,OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)nctx->curn2,OCI_HTYPE_STMT);
      for (i = 0; i < 10; i++)
        OCIHandleFree((dvoid *)(nctx->curn3)[i],OCI_HTYPE_STMT);
      OCIHandleFree((dvoid *)nctx->curn4,OCI_HTYPE_STMT);
      free (nctx);
    }
}

/* the arrays are initialized based on a successful select from */
/* stock/item. We need to shift the values in the orderline array */
/* one position up to compensate when we have an invalid item */

shiftitemstock (i, j)

int i, j;

{
   /* shift up the values for the stock table */
    nctx->s_remote[i] =  nctx->s_remote[j];

   /* shift up the order_line values */

   nctx->nol_i_id_ind[i]=nctx->nol_i_id_ind[j];
   nol_i_id[i] = nol_i_id[j];

   nctx->nol_quantity_ind[i] =  nctx->nol_quantity_ind[j];
   nol_quantity[i] = nol_quantity[j];

   nctx->nol_supply_w_id_ind [i] =  nctx->nol_supply_w_id_ind[j];
   nol_supply_w_id[i] = nol_supply_w_id[j];
}
swapitemstock (i, j)

int i, j;

{

   int k;
   int tempi;
   int tempf;
   char tempstr[52];
   ub2 tempub2;
   sb2 tempsb2;
   OCIRowid  *tmprid;

   tempsb2 = nctx->cons_ind[i];
   nctx->cons_ind[i] = nctx->cons_ind[j];
   nctx->cons_ind[j] = tempsb2;
   tempub2 = nctx->cons_len[i];
   nctx->cons_len[i] = nctx->cons_len[j];
   nctx->cons_len[j] = tempub2;
   tempub2 = nctx->cons_rcode[i];
   nctx->cons_rcode[i] = nctx->cons_rcode[j];
   nctx->cons_rcode[j] = tempub2;
   tempi = nctx->cons[i];
   nctx->cons[i] = nctx->cons[j];
   nctx->cons[j] = tempi;

   tempsb2 = nctx->s_rowid_ind[i];
   nctx->s_rowid_ind[i] = nctx->s_rowid_ind[j];
   nctx->s_rowid_ind[j] = tempsb2;
   tempub2 = nctx->s_rowid_len[i];
   nctx->s_rowid_len[i] = nctx->s_rowid_len[j];
   nctx->s_rowid_len[j] = tempub2;
   tempub2 = nctx->s_rowid_rcode[i];
   nctx->s_rowid_rcode[i] = nctx->s_rowid_rcode[j];
   nctx->s_rowid_rcode[j] = tempub2;
   tmprid = nctx->s_rowid_ptr[i];
   nctx->s_rowid_ptr[i]= nctx->s_rowid_ptr[j];
   nctx->s_rowid_ptr[j]=tmprid;

   tempsb2 = nctx->i_price_ind[i];
   nctx->i_price_ind[i] = nctx->i_price_ind[j];
   nctx->i_price_ind[j] = tempsb2;
   tempub2 = nctx->i_price_len[i];
   nctx->i_price_len[i] = nctx->i_price_len[j];
   nctx->i_price_len[j] = tempub2;
   tempub2 = nctx->i_price_rcode[i];
   nctx->i_price_rcode[i] = nctx->i_price_rcode[j];
   nctx->i_price_rcode[j] = tempub2;
   tempf = i_price[i];
   i_price[i] = i_price[j];
   i_price[j] = tempf;

   tempsb2 = nctx->i_name_ind[i];
   nctx->i_name_ind[i] = nctx->i_name_ind[j];
   nctx->i_name_ind[j] = tempsb2;
   tempub2 = nctx->i_name_len[i];
   nctx->i_name_len[i] = nctx->i_name_len[j];
   nctx->i_name_len[j] = tempub2;
   tempub2 = nctx->i_name_rcode[i];
   nctx->i_name_rcode[i] = nctx->i_name_rcode[j];
   nctx->i_name_rcode[j] = tempub2;
   strncpy (tempstr, i_name[i], 25);
   strncpy (i_name[i], i_name[j], 25);
   strncpy (i_name[j], tempstr, 25);

   tempsb2 = nctx->i_data_ind[i];
   nctx->i_data_ind[i] = nctx->i_data_ind[j];
   nctx->i_data_ind[j] = tempsb2;
   tempub2 = nctx->i_data_len[i];
   nctx->i_data_len[i] = nctx->i_data_len[j];
   nctx->i_data_len[j] = tempub2;
   tempub2 = nctx->i_data_rcode[i];
   nctx->i_data_rcode[i] = nctx->i_data_rcode[j];
   nctx->i_data_rcode[j] = tempub2;
   strncpy (tempstr, nctx->i_data[i], 51);
   strncpy (nctx->i_data[i], nctx->i_data[j], 51);
   strncpy (nctx->i_data[j], tempstr, 51);

   tempsb2 = nctx->s_quantity_ind[i];
   nctx->s_quantity_ind[i] = nctx->s_quantity_ind[j];
   nctx->s_quantity_ind[j] = tempsb2;
   tempub2 = nctx->s_quantity_len[i];
   nctx->s_quantity_len[i] = nctx->s_quantity_len[j];
   nctx->s_quantity_len[j] = tempub2;
   tempub2 = nctx->s_quantity_rcode[i];
   nctx->s_quantity_rcode[i] = nctx->s_quantity_rcode[j];
   nctx->s_quantity_rcode[j] = tempub2;
   tempi = s_quantity[i];
   s_quantity[i] = s_quantity[j];
   s_quantity[j] = tempi;

   tempsb2 = nctx->s_dist_info_ind[i];
   nctx->s_dist_info_ind[i] = nctx->s_dist_info_ind[j];
   nctx->s_dist_info_ind[j] = tempsb2;
   tempub2 = nctx->s_dist_info_len[i];
   nctx->s_dist_info_len[i] = nctx->s_dist_info_len[j];
   nctx->s_dist_info_len[j] = tempub2;
   tempub2 = nctx->s_dist_info_rcode[i];
   nctx->s_dist_info_rcode[i] = nctx->s_dist_info_rcode[j];
   nctx->s_dist_info_rcode[j] = tempub2;
   strncpy (tempstr, nctx->s_dist_info[i], 25);
   strncpy (nctx->s_dist_info[i], nctx->s_dist_info[j], 25);
   strncpy (nctx->s_dist_info[j], tempstr, 25);

   tempsb2 = nctx->s_data_ind[i];
   nctx->s_data_ind[i] = nctx->s_data_ind[j];
   nctx->s_data_ind[j] = tempsb2;
   tempub2 = nctx->s_data_len[i];
   nctx->s_data_len[i] = nctx->s_data_len[j];
   nctx->s_data_len[j] = tempub2;
   tempub2 = nctx->s_data_rcode[i];
   nctx->s_data_rcode[i] = nctx->s_data_rcode[j];
   nctx->s_data_rcode[j] = tempub2;
   strncpy (tempstr, nctx->s_data[i], 51);
   strncpy (nctx->s_data[i], nctx->s_data[j], 51);
   strncpy (nctx->s_data[j], tempstr, 51);

}


SelItemStk ()
{

   int i, j, rpc3,rcount;
```

```
/* array select from item and stock tables */
execstatus=OCIStmtExecute(tpcsvc,(nctx->curn3)[d_id-1],errhp,o_ol_cnt,
                          0,0,0,OCI_DEFAULT);
if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA)) {
  errcode = OCIERROR(errhp,execstatus);
  if(errcode == NOT_SERIALIZABLE) {
     retries++;
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     return (-2);
  } else if (errcode == RECOVERR) {
  /* In case of NO_DATA this should NOT return, but simply fall through */
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     retries++;
     return (-2);
  } else {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     return (-1);
  }
}
/* mark invalid items */
OCIAttrGet((nctx->curn3)[d_id-1], OCI_HTYPE_STMT,&rcount,NULL,
            OCI_ATTR_ROW_COUNT, errhp);
rpc3 = rcount;

/* the result is in order, so we have to shift up to fill    */
/* the slot for the line with the invalid item.              */
/* If more than one item is wrong, this is not an simulated  */
/* error and we'll blow off */

if ((status = o_ol_cnt - rcount) >1)
{
   message ("TPC-C server %d: more than 1 invalid item?\n", proc_no);
  return (rpc3);
}
if (status == 0) return (rpc3);

/* find the invalid item, transfer the rowid information */

for (i = 0; i < o_ol_cnt; i++) {
  if (nctx->cons[i] != i) break;   /* this item is invalid */
}


        message ("TPC-C server %d: reordering items and stocks\n",
                 proc_no);

/* not the last item - shift up */

for (j = i; j < o_ol_cnt-1; j++)
{
  shiftitemstock (j, j+1);
}
/* zero the last item */
  i = o_ol_cnt-1;
  nctx->nol_i_id_ind[i] = NA;
  nctx->nol_supply_w_id_ind[i] = NA;
  nctx->nol_quantity_ind[i] = NA;
  nctx->nol_amount_ind[i] = NA;
  nctx->ol_w_id_ind[i] = NA;
  nctx->ol_d_id_ind[i] = NA;
  nctx->ol_o_id_ind[i] = NA;
  nctx->null_date_ind[i]= NA;
  nctx->ol_number_ind[i] = NA;
  nctx->ol_dist_info_ind[i] = NA;
  nctx->s_remote_ind[i] = NA;
  nctx->s_quant_ind[i] = NA;

  nctx->nol_i_id_len[i] = 0;
  nctx->nol_supply_w_id_len[i] = 0;
  nctx->nol_quantity_len[i] = 0;
  nctx->nol_amount_len[i] = 0;
  nctx->ol_w_id_len[i] = 0;
  nctx->ol_d_id_len[i] = 0;
  nctx->ol_o_id_len[i] = 0;
  nctx->ol_number_len[i] = 0;
  nctx->ol_dist_info_len[i] = 0;
  nctx->null_date_ind[i] = 0;
  nctx->s_remote_len[i] = 0;
    nctx->s_quant_len[i] = 0;

  return (rpc3);

}


UpdStk2 ()
{

  int rpc, rowoff, iters,rcount;

  /* array update of stock table */

  execstatus = OCIStmtExecute(tpcsvc,nctx->curn2,errhp,o_ol_cnt-status,0,0,0,
                              OCI_DEFAULT);
  if(execstatus != OCI_SUCCESS) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    errcode = OCIERROR(errhp,execstatus);
    if(errcode == NOT_SERIALIZABLE) {
       retries++;
         return (-2);
    } else if (errcode == RECOVERR) {
       retries++;
         return (-2);
    } else {
       return -1;
    }
  }
  OCIAttrGet(nctx->curn2,OCI_HTYPE_STMT,&rcount,NULL, OCI_ATTR_ROW_COUNT, errhp);
  rpc = rcount;

  if (rpc != (o_ol_cnt - status)) {
     message ("Error in TPC-C server %d: array update failed\n",
              proc_no);
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     return (-1);
  }

  return (rpc);

}
```

client/oracle/plpay.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: plpay.c 7030100.1 95/07/19 14:44:59 plai Generic<base> $ Copyr (c) 1994
Oracle";
#endif /* RCSID */

/*=================================================================+
  |         Copyright (c) 1995  Oracle Corp, Redwood Shores, CA     |
  |                      OPEN SYSTEMS PERFORMANCE GROUP             |
  |                           All Rights Reserved                   |
  +=================================================================+
  | FILENAME
  |    plpay.c
  | DESCRIPTION
  |    OCI version (using PL/SQL stored procedure) of
  |    PAYMENT transaction in TPC-C benchmark.
  +=================================================================*/

#include "ora_tpcc.h"

#define SQLTXT_INIT "BEGIN initpay.pay_init; END;"
#define SQLTXT_STP  "begin payment.dopayment(:w_id,:d_id,:c_w_id,:c_d_id,     \
                    :c_id,:by_lname,:h_amount,:c_last,:w_street_1,:w_street_2,\
                    :w_city,:w_state,:w_zip,:d_street_1,:d_street_2,:d_city,  \
                    :d_state,:d_zip,:c_first,:c_middle,:c_street_1,           \
                    :c_street_2,:c_city,:c_state,:c_zip,:c_phone,:c_since,    \
                    :c_credit,:c_credit_lim,:c_discount,:c_balance,:c_data,   \
                    :cr_date,:retry); end;"
```

```
struct payctx {
  OCIStmt *curpi;
  OCIStmt *curp0;
  OCIStmt *curp1;
  OCIBind *w_id_bp;
  OCIBind *w_id_bp1;
  sb2 w_id_ind;
  ub2 w_id_len;
  ub2 w_id_rc;

  OCIBind *d_id_bp;
  OCIBind *d_id_bp1;
  sb2 d_id_ind;
  ub2 d_id_len;
  ub2 d_id_rc;

  OCIBind *c_w_id_bp;
  OCIBind *c_w_id_bp1;
  sb2 c_w_id_ind;
  ub2 c_w_id_len;
  ub2 c_w_id_rc;

  OCIBind *c_d_id_bp;
  OCIBind *c_d_id_bp1;
  sb2 c_d_id_ind;
  ub2 c_d_id_len;
  ub2 c_d_id_rc;

  OCIBind *c_id_bp;
  OCIBind *c_id_bp1;
  sb2 c_id_ind;
  ub2 c_id_len;
  ub2 c_id_rc;

  OCIBind *by_lname_bp;

  OCIBind *h_amount_bp;
  OCIBind *h_amount_bp1;
  sb2 h_amount_ind;
  ub2 h_amount_len;
  ub2 h_amount_rc;

  OCIBind *c_last_bp;
  OCIBind *c_last_bp1;
  sb2 c_last_ind;
  ub2 c_last_len;
  ub2 c_last_rc;

  OCIBind *w_street_1_bp;
  OCIBind *w_street_1_bp1;
  sb2 w_street_1_ind;
  ub2 w_street_1_len;
  ub2 w_street_1_rc;

  OCIBind *w_street_2_bp;
  OCIBind *w_street_2_bp1;
  sb2 w_street_2_ind;
  ub2 w_street_2_len;
  ub2 w_street_2_rc;

  OCIBind *w_city_bp;
  OCIBind *w_city_bp1;
  sb2 w_city_ind;
  ub2 w_city_len;
  ub2 w_city_rc;

  OCIBind *w_state_bp;
  OCIBind *w_state_bp1;
  sb2 w_state_ind;
  ub2 w_state_len;
  ub2 w_state_rc;

  OCIBind *w_zip_bp;
  OCIBind *w_zip_bp1;
  sb2 w_zip_ind;
  ub2 w_zip_len;
  ub2 w_zip_rc;

  OCIBind *d_street_1_bp;
  OCIBind *d_street_1_bp1;
  sb2 d_street_1_ind;
  ub2 d_street_1_len;
  ub2 d_street_1_rc;

  OCIBind *d_street_2_bp;
  OCIBind *d_street_2_bp1;
  sb2 d_street_2_ind;
  ub2 d_street_2_len;
  ub2 d_street_2_rc;

  OCIBind *d_city_bp;
  OCIBind *d_city_bp1;
  sb2 d_city_ind;
  ub2 d_city_len;
  ub2 d_city_rc;

  OCIBind *d_state_bp;
  OCIBind *d_state_bp1;
  sb2 d_state_ind;
  ub2 d_state_len;
  ub2 d_state_rc;

  OCIBind *d_zip_bp;
  OCIBind *d_zip_bp1;
  sb2 d_zip_ind;
  ub2 d_zip_len;
  ub2 d_zip_rc;

  OCIBind *c_first_bp;
  OCIBind *c_first_bp1;
  sb2 c_first_ind;
  ub2 c_first_len;
  ub2 c_first_rc;

  OCIBind *c_middle_bp;
  OCIBind *c_middle_bp1;
  sb2 c_middle_ind;
  ub2 c_middle_len;
  ub2 c_middle_rc;

  OCIBind *c_street_1_bp;
  OCIBind *c_street_1_bp1;
  sb2 c_street_1_ind;
  ub2 c_street_1_len;
  ub2 c_street_1_rc;

  OCIBind *c_street_2_bp;
  OCIBind *c_street_2_bp1;
  sb2 c_street_2_ind;
  ub2 c_street_2_len;
  ub2 c_street_2_rc;

  OCIBind *c_city_bp;
  OCIBind *c_city_bp1;
  sb2 c_city_ind;
  ub2 c_city_len;
  ub2 c_city_rc;

  OCIBind *c_state_bp;
  OCIBind *c_state_bp1;
  sb2 c_state_ind;
  ub2 c_state_len;
  ub2 c_state_rc;

  OCIBind *c_zip_bp;
  OCIBind *c_zip_bp1;
  sb2 c_zip_ind;
  ub2 c_zip_len;
  ub2 c_zip_rc;

  OCIBind *c_phone_bp;
  OCIBind *c_phone_bp1;
  sb2 c_phone_ind;
  ub2 c_phone_len;
  ub2 c_phone_rc;

  OCIBind *c_since_bp;
  OCIBind *c_since_bp1;
```

```
  sb2 c_since_ind;
  ub2 c_since_len;
  ub2 c_since_rc;

  OCIBind *c_credit_bp;
  OCIBind *c_credit_bp1;
  sb2 c_credit_ind;
  ub2 c_credit_len;
  ub2 c_credit_rc;

  OCIBind *c_credit_lim_bp;
  OCIBind *c_credit_lim_bp1;
  sb2 c_credit_lim_ind;
  ub2 c_credit_lim_len;
  ub2 c_credit_lim_rc;

  OCIBind *c_discount_bp;
  OCIBind *c_discount_bp1;
  sb2 c_discount_ind;
  ub2 c_discount_len;
  ub2 c_discount_rc;

  OCIBind *c_balance_bp;
  OCIBind *c_balance_bp1;
  sb2 c_balance_ind;
  ub2 c_balance_len;
  ub2 c_balance_rc;

  OCIBind *c_data_bp;
  OCIBind *c_data_bp1;
  sb2 c_data_ind;
  ub2 c_data_len;
  ub2 c_data_rc;

  OCIBind *h_date_bp;
  OCIBind *h_date_bp1;
  sb2 h_date_ind;
  ub2 h_date_len;
  ub2 h_date_rc;

  OCIBind *retries_bp;
  OCIBind *retries_bp1;
  sb2 retries_ind;
  ub2 retries_len;
  ub2 retries_rc;

  OCIBind *cr_date_bp;
  OCIBind *cr_date_bp1;
  sb2 cr_date_ind;
  ub2 cr_date_len;
  ub2 cr_date_rc;

  OCIBind *byln_bp;
  sb2 byln_ind;
  ub2 byln_len;
  ub2 byln_rc;
};

typedef struct payctx payctx;



payctx *pctx;

tkvcpinit ()

{

   text stmbuf[SQL_BUF_SIZE];
   pctx = (payctx *)malloc(sizeof(payctx));
   memset(pctx,(char)0,sizeof(payctx));

/* cursor for init */
   OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx->curpi)),
            OCI_HTYPE_STMT,0,(dvoid**)0));

   OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx->curp0)),
            OCI_HTYPE_STMT,0,(dvoid**)0));

   OCIERROR(errhp,OCIHandleAlloc(tpcenv, (dvoid **)(&(pctx->curp1)),
            OCI_HTYPE_STMT,0,(dvoid**)0));

   /*  build the init statement  and execute it */

   sprintf ((char*)stmbuf, SQLTXT_INIT);
   OCIERROR(errhp,OCIStmtPrepare(pctx->curpi, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));
   OCIERROR(errhp,
     OCIStmtExecute(tpcsvc,pctx->curpi,errhp,1,0,0,0,OCI_DEFAULT));
#ifdef PLSQLPAY
  /*   prepare the stub for calling plsql stored procedure */
   sprintf ((char*)stmbuf, SQLTXT_STP);
   OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));
#else

   /* customer id !=  0, go by last name */

   sqlfile(ORA_BLOCK_PATH "paynz.sql",stmbuf);
   OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

   /* customer id ==  0, go by last name */

   sqlfile(ORA_BLOCK_PATH "payz.sql",stmbuf);
   OCIERROR(errhp,OCIStmtPrepare(pctx->curp1, errhp, stmbuf,
            strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

#endif
  pctx->w_id_ind = TRUE;
  pctx->w_id_len = SIZ(w_id);
  pctx->d_id_ind = TRUE;
  pctx->d_id_len = SIZ(d_id);
  pctx->c_w_id_ind = TRUE;
  pctx->c_w_id_len = SIZ(c_w_id);
  pctx->c_d_id_ind = TRUE;
  pctx->c_d_id_len = SIZ(c_d_id);
  pctx->c_id_ind = TRUE;
  pctx->c_id_len = 0;
  pctx->h_amount_len = SIZ(h_amount);
  pctx->h_amount_ind = TRUE;
  pctx->c_last_ind = TRUE;
  pctx->c_last_len = 0;
  pctx->w_street_1_ind = TRUE;
  pctx->w_street_1_len = 0;
  pctx->w_street_2_ind = TRUE;
  pctx->w_street_2_len = 0;
  pctx->w_city_ind = TRUE;
  pctx->w_city_len = 0;
  pctx->w_state_ind = TRUE;
  pctx->w_state_len = 0;
  pctx->w_zip_ind = TRUE;
  pctx->w_zip_len = 0;
  pctx->d_street_1_ind = TRUE;
  pctx->d_street_1_len = 0;
  pctx->d_street_2_ind = TRUE;
  pctx->d_street_2_len = 0;
  pctx->d_city_ind = TRUE;
  pctx->d_city_len = 0;
  pctx->d_state_ind = TRUE;
  pctx->d_state_len = 0;
  pctx->d_zip_ind = TRUE;
  pctx->d_zip_len = 0;
  pctx->c_first_ind = TRUE;
  pctx->c_first_len = 0;
  pctx->c_middle_ind = TRUE;
  pctx->c_middle_len = 0;
  pctx->c_street_1_ind = TRUE;
  pctx->c_street_1_len = 0;
  pctx->c_street_2_ind = TRUE;
  pctx->c_street_2_len = 0;
  pctx->c_city_ind = TRUE;
  pctx->c_city_len = 0;
  pctx->c_state_ind = TRUE;
  pctx->c_state_len = 0;
  pctx->c_zip_ind = TRUE;
  pctx->c_zip_len = 0;
  pctx->c_phone_ind = TRUE;
```

```
  pctx->c_phone_len = 0;
  pctx->c_since_ind = TRUE;
  pctx->c_since_len = 0;
  pctx->c_credit_ind = TRUE;
  pctx->c_credit_len = 0;
  pctx->c_credit_lim_ind = TRUE;
  pctx->c_credit_lim_len = 0;
  pctx->c_discount_ind = TRUE;
  pctx->c_discount_len = 0;
  pctx->c_balance_ind = TRUE;
  pctx->c_balance_len = sizeof(double);
  pctx->c_data_ind = TRUE;
  pctx->c_data_len = 0;
  pctx->h_date_ind = TRUE;
  pctx->h_date_len = 0;
  pctx->retries_ind = TRUE;
  pctx->retries_len = 0;
  pctx->cr_date_ind = TRUE;
  pctx->cr_date_len = 7;



   /* bind variables */


   OCIBNDR(pctx->curp0, pctx->w_id_bp, errhp,":w_id",ADR(w_id),SIZ(int),
           SQLT_INT, &pctx->w_id_ind, NULL, NULL);
   OCIBNDR(pctx->curp0, pctx->d_id_bp, errhp,":d_id",ADR(d_id),SIZ(int),
           SQLT_INT, &pctx->d_id_ind, NULL, NULL);
   OCIBND(pctx->curp0, pctx->c_w_id_bp, errhp,":c_w_id",ADR(c_w_id),SIZ(int),
           SQLT_INT);
   OCIBND(pctx->curp0, pctx->c_d_id_bp, errhp,":c_d_id",ADR(c_d_id),SIZ(int),
           SQLT_INT);
   OCIBND(pctx->curp0, pctx->c_id_bp, errhp,":c_id",ADR(c_id),SIZ(int),
           SQLT_INT);
#ifdef PLSQLPAY
   OCIBND(pctx->curp0, pctx->by_lname_bp, errhp,":by_lname",ADR(bylastname),
          SIZ(int), SQLT_INT);
#endif
   OCIBNDR(pctx->curp0, pctx->h_amount_bp, errhp,":h_amount",ADR(h_amount),
           SIZ(int),SQLT_INT, &pctx->h_amount_ind, &pctx->h_amount_len,
           &pctx->h_amount_rc);
   OCIBNDR(pctx->curp0, pctx->c_last_bp, errhp,":c_last",c_last,SIZ(c_last),
           SQLT_STR, &pctx->c_last_ind, &pctx->c_last_len, &pctx->c_last_rc);
   OCIBNDR(pctx->curp0, pctx->w_street_1_bp, errhp,":w_street_1",w_street_1,
           SIZ(w_street_1),SQLT_STR, &pctx->w_street_1_ind,
           &pctx->w_street_1_len, &pctx->w_street_1_rc);
   OCIBNDR(pctx->curp0, pctx->w_street_2_bp, errhp,":w_street_2",w_street_2,
           SIZ(w_street_2),SQLT_STR, &pctx->w_street_2_ind,
           &pctx->w_street_2_len, &pctx->w_street_2_rc);
   OCIBNDR(pctx->curp0, pctx->w_city_bp, errhp,":w_city",w_city,SIZ(w_city),
           SQLT_STR, &pctx->w_city_ind, &pctx->w_city_len, &pctx->w_city_rc);
   OCIBNDR(pctx->curp0, pctx->w_state_bp, errhp,":w_state",w_state,SIZ(w_state),
           SQLT_STR, &pctx->w_state_ind, &pctx->w_state_len, &pctx->w_state_rc);
   OCIBNDR(pctx->curp0, pctx->w_zip_bp, errhp,":w_zip",w_zip,SIZ(w_zip),
           SQLT_STR, &pctx->w_zip_ind, &pctx->w_zip_len, &pctx->w_zip_rc);
   OCIBNDR(pctx->curp0, pctx->d_street_1_bp, errhp,":d_street_1",d_street_1,
           SIZ(d_street_1),SQLT_STR, &pctx->d_street_1_ind,
           &pctx->d_street_1_len, &pctx->d_street_1_rc);
   OCIBNDR(pctx->curp0, pctx->d_street_2_bp, errhp,":d_street_2",d_street_2,
           SIZ(d_street_2),SQLT_STR, &pctx->d_street_2_ind,
           &pctx->d_street_2_len, &pctx->d_street_2_rc);
   OCIBNDR(pctx->curp0, pctx->d_city_bp, errhp,":d_city",d_city,SIZ(d_city),
           SQLT_STR, &pctx->d_city_ind, &pctx->d_city_len, &pctx->d_city_rc);
   OCIBNDR(pctx->curp0, pctx->d_state_bp, errhp,":d_state",d_state,SIZ(d_state),
           SQLT_STR, &pctx->d_state_ind, &pctx->d_state_len, &pctx->d_state_rc);
   OCIBNDR(pctx->curp0, pctx->d_zip_bp, errhp,":d_zip",d_zip,SIZ(d_zip),
           SQLT_STR, &pctx->d_zip_ind, &pctx->d_zip_len, &pctx->d_zip_rc);
   OCIBNDR(pctx->curp0, pctx->c_first_bp, errhp,":c_first",c_first,SIZ(c_first),
           SQLT_STR, &pctx->c_first_ind, &pctx->c_first_len, &pctx->c_first_rc);
   OCIBNDR(pctx->curp0, pctx->c_middle_bp, errhp,":c_middle",c_middle,2,
           SQLT_AFC, &pctx->c_middle_ind, &pctx->c_middle_len,
           &pctx->c_middle_rc);
   OCIBNDR(pctx->curp0, pctx->c_street_1_bp, errhp,":c_street_1",c_street_1,
           SIZ(c_street_1),SQLT_STR, &pctx->c_street_1_ind,
           &pctx->c_street_1_len, &pctx->c_street_1_rc);
   OCIBNDR(pctx->curp0, pctx->c_street_2_bp, errhp,":c_street_2",c_street_2,
           SIZ(c_street_2),SQLT_STR, &pctx->c_street_2_ind,
           &pctx->c_street_2_len, &pctx->c_street_2_rc);
   OCIBNDR(pctx->curp0, pctx->c_city_bp, errhp,":c_city",c_city,SIZ(c_city),
           SQLT_STR, &pctx->c_city_ind, &pctx->c_city_len, &pctx->c_city_rc);
   OCIBNDR(pctx->curp0, pctx->c_state_bp, errhp,":c_state",c_state,SIZ(c_state),
           SQLT_STR, &pctx->c_state_ind, &pctx->c_state_len, &pctx->c_state_rc);
   OCIBNDR(pctx->curp0, pctx->c_zip_bp, errhp,":c_zip",c_zip,SIZ(c_zip),
           SQLT_STR, &pctx->c_zip_ind, &pctx->c_zip_len, &pctx->c_zip_rc);
   OCIBNDR(pctx->curp0, pctx->c_phone_bp, errhp,":c_phone",c_phone,SIZ(c_phone),
SQLT_STR, &pctx->c_phone_ind, &pctx->c_phone_len, &pctx->c_phone_rc);
   OCIBNDR(pctx->curp0, pctx->c_since_bp, errhp,":c_since",&c_since,
           SIZ(OCIDate), SQLT_ODT, &pctx->c_since_ind, &pctx->c_since_len,
           &pctx->c_since_rc);
   OCIBNDR(pctx->curp0, pctx->c_credit_bp, errhp,":c_credit",c_credit,
           SIZ(c_credit),SQLT_CHR, &pctx->c_credit_ind, &pctx->c_credit_len,
           &pctx->c_credit_rc);
   OCIBNDR(pctx->curp0, pctx->c_credit_lim_bp, errhp,":c_credit_lim",
           ADR(c_credit_lim),SIZ(int), SQLT_INT, &pctx->c_credit_lim_ind,
           &pctx->c_credit_lim_len, &pctx->c_credit_lim_rc);
   OCIBNDR(pctx->curp0, pctx->c_discount_bp, errhp,":c_discount",
           ADR(c_discount),SIZ(int), SQLT_FLT, &pctx->c_discount_ind,
           &pctx->c_discount_len, &pctx->c_discount_rc);
   OCIBNDR(pctx->curp0, pctx->c_balance_bp, errhp,":c_balance",ADR(c_balance),
           SIZ(double),SQLT_FLT, &pctx->c_balance_ind, &pctx->c_balance_len,
           &pctx->c_balance_rc);
   OCIBNDR(pctx->curp0, pctx->c_data_bp, errhp,":c_data",c_data,SIZ(c_data),
           SQLT_STR, &pctx->c_data_ind, &pctx->c_data_len, &pctx->c_data_rc);
/*
   OCIBNDR(pctx->curp0, pctx->h_date_bp, errhp,":h_date",h_date,SIZ(h_date),
           SQLT_STR, &pctx->h_date_ind, &pctx->h_date_len, &pctx->h_date_rc);
*/
   OCIBNDR(pctx->curp0, pctx->retries_bp, errhp,":retry",ADR(retries),SIZ(int),
           SQLT_INT, &pctx->retries_ind, &pctx->retries_len, &pctx->retries_rc);
   OCIBNDR(pctx->curp0, pctx->cr_date_bp, errhp,":cr_date",ADR(cr_date),
           SIZ(OCIDate),SQLT_ODT, &pctx->cr_date_ind, &pctx->cr_date_len,
           &pctx->cr_date_rc);
#ifndef PLSQLPAY

/* ---- Binds for  the second cursor   */


   OCIBNDR(pctx->curp1, pctx->w_id_bp1, errhp,":w_id",ADR(w_id),SIZ(int),
           SQLT_INT, &pctx->w_id_ind, &pctx->w_id_len, &pctx->w_id_rc);
   OCIBNDR(pctx->curp1, pctx->d_id_bp1, errhp,":d_id",ADR(d_id),SIZ(int),
           SQLT_INT, &pctx->d_id_ind, &pctx->d_id_len, &pctx->d_id_rc);
   OCIBND(pctx->curp1, pctx->c_w_id_bp1, errhp,":c_w_id",ADR(c_w_id),SIZ(int),
           SQLT_INT);
   OCIBND(pctx->curp1, pctx->c_d_id_bp1, errhp,":c_d_id",ADR(c_d_id),SIZ(int),
           SQLT_INT);
   OCIBNDR(pctx->curp1, pctx->c_id_bp1, errhp,":c_id",ADR(c_id),SIZ(int),
           SQLT_INT, &pctx->c_id_ind, &pctx->c_id_len, &pctx->c_id_rc);
   OCIBNDR(pctx->curp1, pctx->h_amount_bp1, errhp,":h_amount",ADR(h_amount),
           SIZ(int),SQLT_INT, &pctx->h_amount_ind, &pctx->h_amount_len,
           &pctx->h_amount_rc);
   OCIBND(pctx->curp1, pctx->c_last_bp1, errhp,":c_last",c_last,SIZ(c_last),
           SQLT_STR);
   OCIBNDR(pctx->curp1, pctx->w_street_1_bp1, errhp,":w_street_1",w_street_1,
           SIZ(w_street_1),SQLT_STR, &pctx->w_street_1_ind,
           &pctx->w_street_1_len, &pctx->w_street_1_rc);
   OCIBNDR(pctx->curp1, pctx->w_street_2_bp1, errhp,":w_street_2",w_street_2,
           SIZ(w_street_2),SQLT_STR, &pctx->w_street_2_ind,
           &pctx->w_street_2_len, &pctx->w_street_2_rc);
   OCIBNDR(pctx->curp1, pctx->w_city_bp1, errhp,":w_city",w_city,SIZ(w_city),
           SQLT_STR, &pctx->w_city_ind, &pctx->w_city_len, &pctx->w_city_rc);
   OCIBNDR(pctx->curp1, pctx->w_state_bp1, errhp,":w_state",w_state,SIZ(w_state),
           SQLT_STR, &pctx->w_state_ind, &pctx->w_state_len, &pctx->w_state_rc);
   OCIBNDR(pctx->curp1, pctx->w_zip_bp1, errhp,":w_zip",w_zip,SIZ(w_zip),
           SQLT_STR, &pctx->w_zip_ind, &pctx->w_zip_len, &pctx->w_zip_rc);
   OCIBNDR(pctx->curp1, pctx->d_street_1_bp1, errhp,":d_street_1",d_street_1,
           SIZ(d_street_1),SQLT_STR, &pctx->d_street_1_ind,
           &pctx->d_street_1_len, &pctx->d_street_1_rc);
   OCIBNDR(pctx->curp1, pctx->d_street_2_bp1, errhp,":d_street_2",d_street_2,
           SIZ(d_street_2),SQLT_STR, &pctx->d_street_2_ind,
           &pctx->d_street_2_len, &pctx->d_street_2_rc);
   OCIBNDR(pctx->curp1, pctx->d_city_bp1, errhp,":d_city",d_city,SIZ(d_city),
           SQLT_STR, &pctx->d_city_ind, &pctx->d_city_len, &pctx->d_city_rc);
   OCIBNDR(pctx->curp1, pctx->d_state_bp1, errhp,":d_state",d_state,
           SIZ(d_state), SQLT_STR, &pctx->d_state_ind, &pctx->d_state_len,
           &pctx->d_state_rc);
   OCIBNDR(pctx->curp1, pctx->d_zip_bp1, errhp,":d_zip",d_zip,SIZ(d_zip),
           SQLT_STR, &pctx->d_zip_ind, &pctx->d_zip_len, &pctx->d_zip_rc);
```

```
   OCIBNDR(pctx->curp1, pctx->c_first_bp1, errhp,":c_first",c_first,
           SIZ(c_first), SQLT_STR, &pctx->c_first_ind, &pctx->c_first_len,
           &pctx->c_first_rc);
   OCIBNDR(pctx->curp1, pctx->c_middle_bp1, errhp,":c_middle",c_middle,2,
           SQLT_AFC, &pctx->c_middle_ind, &pctx->c_middle_len,
           &pctx->c_middle_rc);

   OCIBNDR(pctx->curp1, pctx->c_street_1_bp1, errhp,":c_street_1",c_street_1,
           SIZ(c_street_1),SQLT_STR, &pctx->c_street_1_ind,
           &pctx->c_street_1_len, &pctx->c_street_1_rc);
   OCIBNDR(pctx->curp1, pctx->c_street_2_bp1, errhp,":c_street_2",c_street_2,
           SIZ(c_street_2),SQLT_STR, &pctx->c_street_2_ind,
           &pctx->c_street_2_len, &pctx->c_street_2_rc);
   OCIBNDR(pctx->curp1, pctx->c_city_bp1,
errhp,":c_city",c_city,SIZ(c_city),SQLT_STR,
           &pctx->c_city_ind, &pctx->c_city_len, &pctx->c_city_rc);
   OCIBNDR(pctx->curp1, pctx->c_state_bp1, errhp,":c_state",c_state,SIZ(c_state),
SQLT_STR, &pctx->c_state_ind, &pctx->c_state_len, &pctx->c_state_rc);
   OCIBNDR(pctx->curp1, pctx->c_zip_bp1, errhp,":c_zip",c_zip,SIZ(c_zip),
           SQLT_STR, &pctx->c_zip_ind, &pctx->c_zip_len, &pctx->c_zip_rc);
   OCIBNDR(pctx->curp1, pctx->c_phone_bp1, errhp,":c_phone",c_phone,SIZ(c_phone),
           SQLT_STR, &pctx->c_phone_ind, &pctx->c_phone_len, &pctx->c_phone_rc);
   OCIBNDR(pctx->curp1, pctx->c_since_bp1, errhp,":c_since",&c_since,
           SIZ(OCIDate), SQLT_ODT, &pctx->c_since_ind, &pctx->c_since_len,
           &pctx->c_since_rc);
   OCIBNDR(pctx->curp1, pctx->c_credit_bp1, errhp,":c_credit",c_credit,
           SIZ(c_credit),SQLT_CHR, &pctx->c_credit_ind, &pctx->c_credit_len,
           &pctx->c_credit_rc);
   OCIBNDR(pctx->curp1, pctx->c_credit_lim_bp1, errhp,":c_credit_lim",
           ADR(c_credit_lim),SIZ(int), SQLT_INT, &pctx->c_credit_lim_ind,
           &pctx->c_credit_lim_len, &pctx->c_credit_lim_rc);
   OCIBNDR(pctx->curp1, pctx->c_discount_bp1, errhp,":c_discount",
           ADR(c_discount),SIZ(int), SQLT_FLT, &pctx->c_discount_ind,
           &pctx->c_discount_len, &pctx->c_discount_rc);
   OCIBNDR(pctx->curp1, pctx->c_balance_bp1, errhp,":c_balance",ADR(c_balance),
           SIZ(double),SQLT_FLT, &pctx->c_balance_ind, &pctx->c_balance_len,
           &pctx->c_balance_rc);
   OCIBNDR(pctx->curp1, pctx->c_data_bp1, errhp,":c_data",c_data,SIZ(c_data),
           SQLT_STR, &pctx->c_data_ind, &pctx->c_data_len, &pctx->c_data_rc);
/*
   OCIBNDR(pctx->curp1, pctx->h_date_bp1, errhp,":h_date",h_date,SIZ(h_date),
           SQLT_STR, &pctx->h_date_ind, &pctx->h_date_len, &pctx->h_date_rc);
*/
   OCIBNDR(pctx->curp1, pctx->retries_bp1, errhp,":retry",ADR(retries),SIZ(int),
           SQLT_INT, &pctx->retries_ind, &pctx->retries_len, &pctx->retries_rc);
   OCIBNDR(pctx->curp1, pctx->cr_date_bp1, errhp,":cr_date",ADR(cr_date),
           SIZ(OCIDate),SQLT_ODT, &pctx->cr_date_ind, &pctx->cr_date_len,
           &pctx->cr_date_rc);
#endif

   return (0);

}



tkvcp ()

{

retry:

  pctx->w_id_ind = TRUE;
  pctx->w_id_len = SIZ(w_id);
  pctx->d_id_ind = TRUE;
  pctx->d_id_len = SIZ(d_id);
  pctx->c_w_id_ind = TRUE;
  pctx->c_w_id_len = 0;
  pctx->c_d_id_ind = TRUE;
  pctx->c_d_id_len = 0;
  pctx->c_id_ind = TRUE;
  pctx->c_id_len = 0;
  pctx->h_amount_len = SIZ(h_amount);
  pctx->h_amount_ind = TRUE;
  pctx->c_last_ind = TRUE;
  pctx->c_last_len = SIZ(c_last);
  pctx->w_street_1_ind = TRUE;
  pctx->w_street_1_len = 0;
  pctx->w_street_2_ind = TRUE;
  pctx->w_street_2_len = 0;
  pctx->w_city_ind = TRUE;
  pctx->w_city_len = 0;
  pctx->w_state_ind = TRUE;
  pctx->w_state_len = 0;
  pctx->w_zip_ind = TRUE;
  pctx->w_zip_len = 0;
  pctx->d_street_1_ind = TRUE;
  pctx->d_street_1_len = 0;
  pctx->d_street_2_ind = TRUE;
  pctx->d_street_2_len = 0;
  pctx->d_city_ind = TRUE;
  pctx->d_city_len = 0;
  pctx->d_state_ind = TRUE;
  pctx->d_state_len = 0;
  pctx->d_zip_ind = TRUE;
  pctx->d_zip_len = 0;
  pctx->c_first_ind = TRUE;
  pctx->c_first_len = 0;
  pctx->c_middle_ind = TRUE;
  pctx->c_middle_len = 0;
  pctx->c_street_1_ind = TRUE;
  pctx->c_street_1_len = 0;
  pctx->c_street_2_ind = TRUE;
  pctx->c_street_2_len = 0;
  pctx->c_city_ind = TRUE;
  pctx->c_city_len = 0;
  pctx->c_state_ind = TRUE;
  pctx->c_state_len = 0;
  pctx->c_zip_ind = TRUE;
  pctx->c_zip_len = 0;
  pctx->c_phone_ind = TRUE;
  pctx->c_phone_len = 0;
  pctx->c_since_ind = TRUE;
  pctx->c_since_len = 0;
  pctx->c_credit_ind = TRUE;
  pctx->c_credit_len = 0;
  pctx->c_credit_lim_ind = TRUE;
  pctx->c_credit_lim_len = 0;
  pctx->c_discount_ind = TRUE;
  pctx->c_discount_len = 0;
  pctx->c_balance_ind = TRUE;
  pctx->c_balance_len = sizeof(double);
  pctx->c_data_ind = TRUE;
  pctx->c_data_len = 0;
  pctx->h_date_ind = TRUE;
  pctx->h_date_len = 0;
  pctx->retries_ind = TRUE;
  pctx->retries_len = 0;
  pctx->cr_date_ind = TRUE;
  pctx->cr_date_len = 7;

#ifdef PLSQLPAY
     execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp0,errhp,1,0,0,0,OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
#else
  if(bylastname) {
    execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp1,errhp,1,0,0,0,OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
  } else {
    execstatus=OCIStmtExecute(tpcsvc,pctx-
>curp0,errhp,1,0,0,0,OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
  }
#endif

  if(execstatus != OCI_SUCCESS) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }
   return 0;
```

```
}


void tkvcpdone ()

{
  if(pctx) {
    free(pctx);
  }
}
```

client/oracle/plord.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: plord.c 7030100.1 95/07/19 14:46:13 plai Generic<base> $ Copyr (c) 1994
Oracle";
#endif /* RCSID */

/*============================================================================+
 |         Copyright (c) 1995  Oracle Corp, Redwood Shores, CA                |
 |                      OPEN SYSTEMS PERFORMANCE GROUP                        |
 |                         All Rights Reserved                                |
 +============================================================================+
 | FILENAME
 |    plord.c
 | DESCRIPTION
 |    OCI version (using PL/SQL anynomous block) of
 |    ORDER STATUS transaction in TPC-C benchmark.
 +============================================================================*/

#include "ora_tpcc.h"

#ifdef PLSQLORD
#define SQLTXT "BEGIN orderstatus.getstatus (:w_id, :d_id, :c_id, :byln, \
   :c_last, :c_first, :c_middle, :c_balance, :o_id, :o_entry_d, :o_cr_id, \
   :o_ol_cnt, :ol_s_w_id, :ol_i_id, :ol_quantity, :ol_amount, :ol_d_d); END;"
#else

#define SQLCUR0 "SELECT rowid FROM customer \
                 WHERE c_d_id = :d_id AND c_w_id = :w_id AND c_last = :c_last \
                 ORDER BY c_last, c_d_id, c_w_id, c_first"

#define SQLCUR1 "SELECT c_id, c_balance, c_first, c_middle, c_last, \
                 o_id, o_entry_d, o_carrier_id, o_ol_cnt \
                 FROM customer, orders \
                 WHERE customer.rowid = :cust_rowid \
                 AND   o_d_id = c_d_id AND o_w_id = c_w_id AND o_c_id = c_id \
                 ORDER BY o_c_id, o_d_id, o_w_id, o_id DESC"


#define SQLCUR2 "SELECT c_balance, c_first, c_middle, c_last, \
                 o_id, o_entry_d, o_carrier_id, o_ol_cnt,c_id \
                 FROM customer, orders \
                 WHERE c_id = :c_id AND c_d_id = :d_id AND c_w_id = :w_id \
                 AND o_d_id = c_d_id AND o_w_id = c_w_id AND o_c_id = c_id \
                 ORDER BY o_c_id, o_d_id, o_w_id, o_id DESC"


#define SQLCUR3 "SELECT ol_i_id,ol_supply_w_id,ol_quantity,ol_amount, \
                ol_delivery_d\
                FROM order_line \
                WHERE ol_d_id = :d_id AND ol_w_id = :w_id AND ol_o_id = :o_id"

#define SQLCUR4 "SELECT count(c_last) FROM customer \
                 WHERE c_d_id = :d_id AND c_w_id = :w_id AND c_last = :c_last "
#endif



struct ordctx {
   sb2 c_rowid_ind[100];
   sb2 ol_supply_w_id_ind[NITEMS];
   sb2 ol_i_id_ind[NITEMS];
   sb2 ol_quantity_ind[NITEMS];
   sb2 ol_amount_ind[NITEMS];
   sb2 ol_delivery_d_ind[NITEMS];
   sb2 ol_w_id_ind;
   sb2 ol_d_id_ind;
   sb2 ol_o_id_ind;
   sb2 c_id_ind;
   sb2 c_first_ind;
   sb2 c_middle_ind;
   sb2 c_balance_ind;
   sb2 c_last_ind;
   sb2 o_id_ind;
   sb2 o_entry_d_ind;
   sb2 o_carrier_id_ind;
   sb2 o_ol_cnt_ind;

   ub4 c_rowid_len[100];
   ub2 ol_supply_w_id_len[NITEMS];
   ub2 ol_i_id_len[NITEMS];
   ub2 ol_quantity_len[NITEMS];
   ub2 ol_amount_len[NITEMS];
   ub2 ol_delivery_d_len[NITEMS];
   ub2 ol_w_id_len;
   ub2 ol_d_id_len;
   ub2 ol_o_id_len;

   ub2 c_rowid_rcode[100];
   ub2 ol_supply_w_id_rcode[NITEMS];
   ub2 ol_i_id_rcode[NITEMS];
   ub2 ol_quantity_rcode[NITEMS];
   ub2 ol_amount_rcode[NITEMS];
   ub2 ol_delivery_d_rcode[NITEMS];
   ub2 ol_w_id_rcode;
   ub2 ol_d_id_rcode;
   ub2 ol_o_id_rcode;

   ub4 ol_supply_w_id_csize;
   ub4 ol_i_id_csize;
   ub4 ol_quantity_csize;
   ub4 ol_amount_csize;
   ub4 ol_delivery_d_csize;
   ub4 ol_w_id_csize;
   ub4 ol_d_id_csize;
   ub4 ol_o_id_csize;

   OCIStmt *curo0;
   OCIBind *w_id_bp0;
   OCIBind *d_id_bp0;
   OCIBind *c_id_bp;
   OCIBind *c_last_bp;
#ifdef PLSQLORD
   OCIBind *byln_bp;
   OCIBind *c_first_bp;
   OCIBind *c_middle_bp;
   OCIBind *c_balance_bp;
   OCIBind *o_entry_d_bp;
   OCIBind *o_cr_id_bp;
   OCIBind *o_ol_cnt_bp;
   OCIBind *ol_i_id_bp;
   OCIBind *ol_supply_w_id_bp;
   OCIBind *ol_quantity_bp;
   OCIBind *ol_amount_bp;
   OCIBind *ol_d_base_bp;
   ub4 ol_i_id_cnt;
   ub4 ol_sup_cnt;
   ub4 ol_qty_cnt;
   ub4 ol_amt_cnt;
   ub4 ol_del_d_cnt;
#else
   OCIStmt *curo1;
   OCIStmt *curo2;
   OCIStmt *curo3;
   OCIStmt *curo4;
   OCIBind *w_id_bp2;
   OCIBind *w_id_bp3;
   OCIBind *w_id_bp4;
   OCIBind *d_id_bp2;
   OCIBind *d_id_bp3;
   OCIBind *d_id_bp4;
   OCIBind *c_last_bp4;
```

```
   OCIBind *o_id_bp;
   OCIBind *c_rowid_bp;
   OCIDefine *c_rowid_dp;
   OCIDefine *c_last_dp;
   OCIDefine *c_last_dp1;
   OCIDefine *c_id_dp;
   OCIDefine *c_id_dp1;
   OCIDefine *c_first_dp1;
   OCIDefine *c_first_dp2;
   OCIDefine *c_middle_dp1;
   OCIDefine *c_middle_dp2;
   OCIDefine *c_balance_dp1;
   OCIDefine *c_balance_dp2;
   OCIDefine *o_id_dp1;
   OCIDefine *o_id_dp2;
   OCIDefine *o_entry_d_dp1;
   OCIDefine *o_entry_d_dp2;
   OCIDefine *o_cr_id_dp1;
   OCIDefine *o_cr_id_dp2;
   OCIDefine *o_ol_cnt_dp1;
   OCIDefine *o_ol_cnt_dp2;
   OCIDefine *ol_d_d_dp;
   OCIDefine *ol_i_id_dp;
   OCIDefine *ol_supply_w_id_dp;
   OCIDefine *ol_quantity_dp;
   OCIDefine *ol_amount_dp;
   OCIDefine *ol_d_base_dp;
   OCIDefine *c_count_dp;
   OCIRowid *c_rowid_ptr[100];
   OCIRowid *middle_cust;

   int cs;
   int cust_idx;
   int norow;
   int rcount;
   int somerows;
#endif
};

typedef struct ordctx ordctx;

struct defctx
{
 boolean reexec;
 ub4 count;
};
typedef struct defctx defctx;

ordctx *octx;

defctx cbctx;

#ifndef PLSQLORD
sb4 rid_data(dvoid *ctxp, OCIDefine *dp, ub4 iter,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  ub4 i;
  if (((defctx*)ctxp)->reexec)/* if this is the second execute - use entry 0 */
  {
     i = 0;
     ((defctx*)ctxp)->count--;   /* count down */
  }
  else
     i = iter;
  *bufpp = octx->c_rowid_ptr[i];
  *indpp= &octx->c_rowid_ind[i];
  *alenp= &octx->c_rowid_len[i];
  *rcodepp = &octx->c_rowid_rcode[i];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
#endif

tkvcoinit ()

{

   int i;
   text stmbuf[SQL_BUF_SIZE];

   octx = (ordctx *) malloc (sizeof(ordctx));
   memset(octx,(char)0,sizeof(ordctx));
#ifndef PLSQLORD
   octx->cs = 1;
   octx->norow = 0;
   octx->somerows = 10;
/* get the rowid handles */
   for(i=0;i<100;i++) {
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&octx->c_rowid_ptr[i],
      OCI_DTYPE_ROWID,0,(dvoid**)0));
  }
#endif


   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
#ifndef PLSQLORD
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo1,OCI_HTYPE_STMT,0,(dvoid**)0));
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo2,OCI_HTYPE_STMT,0,(dvoid**)0));
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo3,OCI_HTYPE_STMT,0,(dvoid**)0));
   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo4,OCI_HTYPE_STMT,0,(dvoid**)0));
#endif

#ifdef PLSQLORD
   sprintf((char *) stmbuf, SQLTXT);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo0,errhp,stmbuf,strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX,OCI_DEFAULT));
#else
/*  c_id = 0, use find customer by lastname. Get an array or rowid's back*/
   sprintf((char *) stmbuf, SQLCUR0);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo0,errhp,stmbuf,strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX,OCI_DEFAULT));
   OCIERROR(errhp,
      OCIAttrSet(octx->curo0,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                  OCI_ATTR_PREFETCH_ROWS,errhp));
/* get order/customer info back based on rowid */
   sprintf((char *) stmbuf, SQLCUR1);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo1,errhp,stmbuf,strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX,OCI_DEFAULT));
   OCIERROR(errhp,
      OCIAttrSet(octx->curo1,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                  OCI_ATTR_PREFETCH_ROWS,errhp));

/*  c_id == 0, use lastname to find customer */
   sprintf((char *) stmbuf, SQLCUR2);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo2,errhp,stmbuf,strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX,OCI_DEFAULT));
   OCIERROR(errhp,
      OCIAttrSet(octx->curo2,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                  OCI_ATTR_PREFETCH_ROWS,errhp));

   sprintf((char *) stmbuf, SQLCUR3);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo3,errhp,stmbuf,strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX,OCI_DEFAULT));
   OCIERROR(errhp,
      OCIAttrSet(octx->curo3,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                  OCI_ATTR_PREFETCH_ROWS,errhp));

   sprintf((char *) stmbuf, SQLCUR4);
   OCIERROR(errhp,
      OCIStmtPrepare(octx->curo4,errhp,stmbuf,strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX,OCI_DEFAULT));
   OCIERROR(errhp,
      OCIAttrSet(octx->curo4,OCI_HTYPE_STMT,(dvoid*)&octx->norow,0,
                  OCI_ATTR_PREFETCH_ROWS,errhp));
#endif
```

```
   for (i = 0; i < NITEMS; i++) {
      octx->ol_supply_w_id_ind[i] = TRUE;
      octx->ol_i_id_ind[i] = TRUE;
      octx->ol_quantity_ind[i] = TRUE;
      octx->ol_amount_ind[i] = TRUE;
      octx->ol_delivery_d_ind[i] = TRUE;

      octx->ol_supply_w_id_len[i] = sizeof(int);
      octx->ol_i_id_len[i] = sizeof(int);
      octx->ol_quantity_len[i] = sizeof(int);
      octx->ol_amount_len[i] = sizeof(int);
      octx->ol_delivery_d_len[i] = sizeof(ol_d_base[0]);
   }
   octx->ol_supply_w_id_csize = NITEMS;
   octx->ol_i_id_csize = NITEMS;
   octx->ol_quantity_csize = NITEMS;
   octx->ol_amount_csize = NITEMS;
   octx->ol_delivery_d_csize = NITEMS;
   octx->ol_w_id_csize = NITEMS;
   octx->ol_o_id_csize = NITEMS;
   octx->ol_d_id_csize = NITEMS;
   octx->ol_w_id_ind = TRUE;
   octx->ol_d_id_ind = TRUE;
   octx->ol_o_id_ind = TRUE;
   octx->ol_w_id_len = sizeof(int);
   octx->ol_d_id_len = sizeof(int);
   octx->ol_o_id_len = sizeof(int);

   /* bind variables */
#ifdef PLSQLORD

   OCIBND(octx->curo0, octx->w_id_bp0, errhp,":w_id",ADR(w_id),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo0, octx->d_id_bp0, errhp,":d_id",ADR(d_id),
          SIZ(int), SQLT_INT);
   OCIBND(octx->curo0, octx->c_id_bp , errhp,":c_id",ADR(c_id),
          SIZ(c_id),SQLT_INT);
   OCIBND(octx->curo0, octx->byln_bp , errhp,":byln",ADR(bylastname),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo0, octx->c_last_bp , errhp,":c_last",c_last,
          SIZ(c_last),SQLT_STR);
   OCIBND(octx->curo0, octx->c_first_bp , errhp,":c_first",c_first,
          SIZ(c_first),SQLT_STR);
   OCIBND(octx->curo0, octx->c_middle_bp , errhp,":c_middle",c_middle,
          SIZ(c_middle),SQLT_STR);
   OCIBND(octx->curo0, octx->c_balance_bp , errhp,":c_balance",
          ADR(c_balance),SIZ (float),SQLT_FLT);
   OCIBND(octx->curo0, octx->c_id_bp , errhp,":o_id",ADR(o_id),
          SIZ(int),SQLT_INT);
   OCIBND(octx->curo0, octx->o_entry_d_bp , errhp,":o_entry_d",o_entry_d,
          SIZ(o_entry_d),SQLT_STR);
   OCIBND(octx->curo0, octx->o_cr_id_bp , errhp,":o_cr_id",ADR(o_carrier_id),
          SIZ(int), SQLT_INT);
   OCIBND(octx->curo0, octx->o_ol_cnt_bp , errhp,":o_ol_cnt",ADR(o_ol_cnt),
          SIZ(int),SQLT_INT);

   OCIBNDRAA(octx->curo0, octx->ol_i_id_bp, errhp, ":ol_i_id",
            ol_i_id,SIZ(int),SQLT_INT,
            octx->ol_i_id_ind,octx->ol_i_id_len,
            octx->ol_i_id_rcode,NITEMS,&octx->ol_i_id_cnt);
   OCIBNDRAA(octx->curo0,octx->ol_supply_w_id_bp,errhp,":ol_s_w_id",
            ol_supply_w_id,SIZ(int),SQLT_INT,
            octx->ol_supply_w_id_ind,octx->ol_supply_w_id_len,
            octx->ol_supply_w_id_rcode,NITEMS,&octx->ol_sup_cnt);
   OCIBNDRAA(octx->curo0, octx->ol_quantity_bp,errhp,":ol_quantity",
            ol_quantity,SIZ(int),SQLT_INT,
            octx->ol_quantity_ind,octx->ol_quantity_len,
            octx->ol_quantity_rcode,NITEMS,&octx->ol_qty_cnt);
   OCIBNDRAA(octx->curo0,octx->ol_amount_bp,errhp,":ol_amount",ol_amount,
            SIZ(float),SQLT_FLT,octx->ol_amount_ind,
            octx->ol_amount_len, octx->ol_amount_rcode,NITEMS,
            &octx->ol_amt_cnt);
   OCIBNDRAA(octx->curo0,octx->ol_d_base_bp,errhp,":ol_d_d",ol_d_base,
            SIZ(OCIDate),SQLT_ODT,octx->ol_delivery_d_ind,
            octx->ol_delivery_d_len, octx->ol_delivery_d_rcode,NITEMS,
            &octx->ol_del_d_cnt);

#else
   /* c_id (customer id) is not known */
   OCIBND(octx->curo0,octx->w_id_bp0,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo0,octx->d_id_bp0,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo0,octx->c_last_bp,errhp,":c_last",c_last,SIZ(c_last),
          SQLT_STR);
   OCIDFNDYN(octx->curo0,octx->c_rowid_dp,errhp,1,octx->c_rowid_ptr,
           SIZ(OCIRowid*), SQLT_RDD, octx->c_rowid_ind, &cbctx,rid_data);

   OCIBND(octx->curo1,octx->c_rowid_bp,errhp,":cust_rowid",
         &octx->middle_cust, sizeof( octx->middle_cust),SQLT_RDD);
   OCIDEF(octx->curo1,octx->c_id_dp,errhp,1,ADR(c_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->c_balance_dp1,errhp,2,ADR(c_balance),
          SIZ(double),SQLT_FLT);
   OCIDEF(octx->curo1,octx->c_first_dp1,errhp,3,c_first,SIZ(c_first)-1,
          SQLT_CHR);
   OCIDEF(octx->curo1,octx->c_middle_dp1,errhp,4,c_middle,
          SIZ(c_middle)-1,SQLT_AFC);
   OCIDEF(octx->curo1,octx->c_last_dp1,errhp,5,c_last,SIZ(c_last)-1,
          SQLT_CHR);
   OCIDEF(octx->curo1,octx->o_id_dp1,errhp,6,ADR(o_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_entry_d_dp1,errhp,7,
          &o_entry_d_base,SIZ(OCIDate),SQLT_ODT);
   OCIDEF(octx->curo1,octx->o_cr_id_dp1,errhp,8,ADR(o_carrier_id),
          SIZ(int),SQLT_INT);
   OCIDEF(octx->curo1,octx->o_ol_cnt_dp1,errhp,9,ADR(o_ol_cnt),
          SIZ(int),SQLT_INT);

/* Bind for third cursor , no-zero customer id */
   OCIBND(octx->curo2,octx->w_id_bp2,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo2,octx->d_id_bp2,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo2,octx->c_id_bp,errhp,":c_id",ADR(c_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo2,octx->c_balance_dp2,errhp,1,ADR(c_balance),
          SIZ(double),SQLT_FLT);
   OCIDEF(octx->curo2,octx->c_first_dp2,errhp,2,c_first,SIZ(c_first)-1,
          SQLT_CHR);
   OCIDEF(octx->curo2,octx->c_middle_dp2,errhp,3,c_middle,
          SIZ(c_middle)-1,SQLT_AFC);
   OCIDEF(octx->curo2,octx->c_last_dp,errhp,4,c_last,SIZ(c_last)-1, SQLT_CHR);
   OCIDEF(octx->curo2,octx->o_id_dp2,errhp,5,ADR(o_id),SIZ(int),SQLT_INT);
   OCIDEF(octx->curo2,octx->o_entry_d_dp2,errhp,6, &o_entry_d_base,
          SIZ(OCIDate),SQLT_ODT);
   OCIDEF(octx->curo2, octx->o_cr_id_dp2,errhp,7,ADR(o_carrier_id),
          SIZ(int), SQLT_INT);
   OCIDEF(octx->curo2,octx->o_ol_cnt_dp2,errhp,8,ADR(o_ol_cnt),
          SIZ(int),SQLT_INT);
   OCIDEF(octx->curo2,octx->c_id_dp1,errhp,9,ADR(c_id),SIZ(int),SQLT_INT);

/* Bind for last cursor */

   OCIBND(octx->curo3,octx->w_id_bp3,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo3,octx->d_id_bp3,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo3,octx->o_id_bp,errhp,":o_id",ADR(o_id),SIZ(int),SQLT_INT);

   OCIDFNRA(octx->curo3, octx->ol_i_id_dp, errhp, 1, ol_i_id,SIZ(int),SQLT_INT,
            octx->ol_i_id_ind,octx->ol_i_id_len, octx->ol_i_id_rcode);
   OCIDFNRA(octx->curo3,octx->ol_supply_w_id_dp,errhp,2, ol_supply_w_id,
           SIZ(int),SQLT_INT, octx->ol_supply_w_id_ind,
           octx->ol_supply_w_id_len, octx->ol_supply_w_id_rcode);
   OCIDFNRA(octx->curo3, octx->ol_quantity_dp,errhp,3, ol_quantity,SIZ(int),
           SQLT_INT, octx->ol_quantity_ind,octx->ol_quantity_len,
             octx->ol_quantity_rcode);
   OCIDFNRA(octx->curo3,octx->ol_amount_dp,errhp,4,ol_amount, SIZ(int),
           SQLT_INT,octx->ol_amount_ind, octx->ol_amount_len,
           octx->ol_amount_rcode);
   OCIDFNRA(octx->curo3,octx->ol_d_base_dp,errhp,5,ol_d_base,SIZ(OCIDate),
           SQLT_ODT, octx->ol_delivery_d_ind,octx->ol_delivery_d_len,
             octx->ol_delivery_d_rcode);

   OCIBND(octx->curo4,octx->w_id_bp4,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo4,octx->d_id_bp4,errhp,":d_id",ADR(d_id),SIZ(int),SQLT_INT);
   OCIBND(octx->curo4,octx->c_last_bp4,errhp,":c_last",c_last,SIZ(c_last),
          SQLT_STR);
   OCIDEF(octx->curo4,octx->c_count_dp,errhp,1,ADR(octx->rcount),SIZ(int),
          SQLT_INT);


#endif
   return (0);
```

```
}


tkvco ()

{

   int i;
   int rcount;

   for (i = 0; i < NITEMS; i++) {
      octx->ol_supply_w_id_ind[i] = TRUE;
      octx->ol_i_id_ind[i] = TRUE;
      octx->ol_quantity_ind[i] = TRUE;
      octx->ol_amount_ind[i] = TRUE;
      octx->ol_delivery_d_ind[i] = TRUE;
      octx->ol_supply_w_id_len[i] = sizeof(int);
      octx->ol_i_id_len[i] = sizeof(int);
      octx->ol_quantity_len[i] = sizeof(int);
      octx->ol_amount_len[i] = sizeof(int);
      octx->ol_delivery_d_len[i] = sizeof(OCIDate);
   }
   octx->ol_supply_w_id_csize = NITEMS;
   octx->ol_i_id_csize = NITEMS;
   octx->ol_quantity_csize = NITEMS;
   octx->ol_amount_csize = NITEMS;
   octx->ol_delivery_d_csize = NITEMS;
#ifdef PLSQLORD
   octx->ol_i_id_cnt = 0;
   octx->ol_sup_cnt = 0;
   octx->ol_qty_cnt = 0;
   octx->ol_amt_cnt = 0;
   octx->ol_del_d_cnt = 0;
   OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,octx->curo0,errhp,1,0,0,0,OCI_DEFAULT));
#else
retry:
   if(bylastname)
   {
     cbctx.reexec = FALSE;
     execstatus=OCIStmtExecute(tpcsvc,octx->curo0,errhp,100,0,0,0,OCI_DEFAULT);
     /* will get OCI_NO_DATA if <100 found */
     if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
     {
       errcode=OCIERROR(errhp, execstatus);
       if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
       {
         OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
         retries++;
         goto retry;
       } else {
         return -1;
       }
     }
     if (execstatus == OCI_NO_DATA) /* there are no more rows */
     {
      /* get rowcount, find middle one */
       OCIAttrGet(octx->curo0,OCI_HTYPE_STMT,&rcount,NULL,OCI_ATTR_ROW_COUNT,errhp);
       if (rcount <1)
       {
         message("No Data Found\n");
         return (-1);
       }
       octx->cust_idx=(rcount-1)/2 ;
     }
     else
     {
       /* count the number of rows */
       execstatus=OCIStmtExecute(tpcsvc,octx->curo4,errhp,1,0,0,0,OCI_DEFAULT);
       if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
       {
         if ((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
         {
           OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
           retries++;
           goto retry;
         } else {
           return -1;
         }
       }
       if (octx->rcount+1 < 200 )
          octx->cust_idx=(octx->rcount-1)/2 ;
       else            /* */
       {
          cbctx.reexec = TRUE;
          cbctx.count = (octx->rcount+1)/2 ;
          execstatus=OCIStmtExecute(tpcsvc,octx->curo0,errhp,cbctx.count,
                                    0,0,0,OCI_DEFAULT);
          /* will get OCI_NO_DATA if <100 found */
          if (cbctx.count > 0)
          {
             message ("did not get all rows ");
             return (-1);
          }

          if ((execstatus != OCI_NO_DATA) && (execstatus != OCI_SUCCESS))
          {
            errcode=OCIERROR(errhp, execstatus);
            if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
            {
              OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
              retries++;
              goto retry;
             } else {
             return -1;
             }
          }
          octx->cust_idx=0 ;
       }
   }

   octx->middle_cust = octx->c_rowid_ptr[octx->cust_idx];
   execstatus=OCIStmtExecute(tpcsvc,octx->curo1,errhp,1,0,0,0,OCI_DEFAULT);
   if (execstatus != OCI_SUCCESS)
   {
      errcode=OCIERROR(errhp,execstatus);
      OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
      if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
      {
         retries++;
         goto retry;
         } else {
         return -1;
      }
   }
  }
}
else
{
  execstatus=OCIStmtExecute(tpcsvc,octx->curo2,errhp,1,0,0,0,OCI_DEFAULT);
  if (execstatus != OCI_SUCCESS)
  {
    errcode=OCIERROR(errhp,execstatus);
    OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
    if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
    {
      retries++;
      goto retry;
    }
    else
    {
      return -1;
    }
  }
}
octx->ol_w_id_ind = TRUE;
octx->ol_d_id_ind = TRUE;
octx->ol_o_id_ind = TRUE;
octx->ol_w_id_len = sizeof(int);
octx->ol_d_id_len = sizeof(int);
octx->ol_o_id_len = sizeof(int);

execstatus = OCIStmtExecute(tpcsvc,octx->curo3,errhp,o_ol_cnt,0,0,0,
                  OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
if (execstatus != OCI_SUCCESS )
{
  errcode=OCIERROR(errhp,execstatus);
```

```
      OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
      if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
      {
        retries++;
        goto retry;
      }
      else
      {
        return -1;
      }

    }
#endif
   /* clean up and convert the delivery dates */
    for (i = 0; i < o_ol_cnt; i++)
    {
      if (octx->ol_delivery_d_ind[i] == -1) /* null date in field */
          strncpy((char*)ol_delivery_d[i],ORA_NULL_DATE,10);
      else
      {
        ol_del_len[i]=sizeof(ol_delivery_d[i]);
        OCIERROR(errhp,OCIDateToText(errhp,&ol_d_base[i],

(text*)SHORTDATE,strlen(SHORTDATE),(text*)0,0,&ol_del_len[i],ol_delivery_d[i]));
      }
    }

   return (0);

}



void tkvcodone ()

{

   if (octx)
      free (octx);

}
```

client/oracle/plsto.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: plsto.c 7010000.3 95/02/14 12:48:03 plai Generic<base> $ Copyr (c) 1994
Oracle";
#endif /* RCSID */

/*============================================================================+
 |         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA                |
 |                     OPEN SYSTEMS PERFORMANCE GROUP                         |
 |                          All Rights Reserved                               |
 +============================================================================+
 | FILENAME
 |    plsto.c
 | DESCRIPTION
 |    OCI version of STOCK LEVEL transaction in TPC-C benchmark.
 +============================================================================*/

#include "ora_tpcc.h"

#ifdef PLSQLSTO
#define SQLTXT "BEGIN stocklevel.getstocklevel (:w_id, :d_id, :threshold, \
   :low_stock); END;"
#else
#define SQLTXT "SELECT /*+ nocache(stock) */ \
               count (DISTINCT s_i_id) \
            FROM order_line, stock, district \
            WHERE d_id = :d_id AND d_w_id = :w_id AND \
                  d_id = ol_d_id AND d_w_id = ol_w_id AND \
                  ol_i_id = s_i_id AND ol_w_id = s_w_id AND \
                  s_quantity < :threshold AND \
                  ol_o_id BETWEEN (d_next_o_id - 20) AND (d_next_o_id - 1)"
#endif

struct stoctx {
  OCIStmt *curs;
  OCIBind *w_id_bp;
  OCIBind *d_id_bp;
  OCIBind *threshold_bp;
#ifdef PLSQLSTO
  OCIBind *low_stock_bp;
#else
  OCIDefine *low_stock_bp;
#endif
  int norow;
};

typedef struct stoctx stoctx;

stoctx *sctx;


tkvcsinit ()

{

   text stmbuf[SQL_BUF_SIZE];
   sctx = (stoctx *)malloc(sizeof(stoctx));
   memset(sctx,(char)0,sizeof(stoctx));

   sctx->norow=0;

   OCIERROR(errhp,
      OCIHandleAlloc(tpcenv,(dvoid**)&sctx->curs,OCI_HTYPE_STMT,0,(dvoid**)0));
   sprintf ((char *) stmbuf, SQLTXT);
   OCIERROR(errhp,OCIStmtPrepare(sctx->curs,errhp,stmbuf,strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX,OCI_DEFAULT));
#ifndef PLSQLSTO
   OCIERROR(errhp,
      OCIAttrSet(sctx->curs,OCI_HTYPE_STMT,(dvoid*)&sctx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));
#endif

   /* bind variables */

   OCIBND(sctx->curs,sctx->w_id_bp,errhp, ":w_id", ADR(w_id),sizeof(int),
          SQLT_INT);
   OCIBND(sctx->curs,sctx->d_id_bp,errhp, ":d_id", ADR(d_id),sizeof(int),
      SQLT_INT);
   OCIBND(sctx->curs,sctx->threshold_bp,errhp, ":threshold", ADR(threshold),
      sizeof(int),SQLT_INT);
#ifdef PLSQLSTO
   OCIBND(sctx->curs,sctx->low_stock_bp,errhp,":low_stock" , ADR(low_stock),
      sizeof(int), SQLT_INT);
#else
   OCIDEFINE(sctx->curs,sctx->low_stock_bp,errhp, 1, ADR(low_stock),
      sizeof(int), SQLT_INT);
#endif

   return (0);

}



tkvcs ()

{

retry:
    execstatus= OCIStmtExecute(tpcsvc,sctx->curs,errhp,1,0,0,0,
                               OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);
    if (execstatus != OCI_SUCCESS)
    {
      errcode=OCIERROR(errhp,execstatus);
      OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
      if((errcode == NOT_SERIALIZABLE) || (errcode == RECOVERR))
      {
         retries++;
         goto retry;
      } else {
         return -1;
```

```
      }
     }

  return (0);
}


void tkvcsdone ()

{
  if(sctx) free(sctx);
}
```

client/oracle/pldel.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: pldel.c 7030100.5 96/06/24 16:26:06 plai Generic<base> $ Copyr (c) 1994
Oracle";
#endif /* RCSID */

/*============================================================================+
 |        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA                 |
 |                     OPEN SYSTEMS PERFORMANCE GROUP                         |
 |                          All Rights Reserved                               |
 +============================================================================+
 | FILENAME
 |    pldel.c
 | DESCRIPTION
 |    OCI version of DELIVERY transaction in TPC-C benchmark.
 +============================================================================*/

#include "ora_tpcc.h"

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
#define SQLTXT0 "SELECT substr(value,1,5) FROM v$parameter \
   WHERE name = 'instance_number'"
#endif

#ifdef PLSQLDEL
#define SQLTXT "BEGIN delivery.deliver (:w_id, :carrier_id, :order_id,\
 :retry); END;"
#else
# ifdef DMLRETDEL
#define SQLTXT1 "DELETE FROM new_order WHERE no_d_id = :d_id \
                 AND no_w_id = :w_id and rownum <= 1 \
                RETURNING no_o_id into :o_id "
# else
#define SQLTXT1A "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 1, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 1 AND o_w_id = :w_id AND o_d_id = 1 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1B "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 2, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 2 AND o_w_id = :w_id AND o_d_id = 2 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1C "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 3, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 3 AND o_w_id = :w_id AND o_d_id = 3 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1D "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 4, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 4 AND o_w_id = :w_id AND o_d_id = 4 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1E "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 5, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 5 AND o_w_id = :w_id AND o_d_id = 5 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1F "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 6, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 6 AND o_w_id = :w_id AND o_d_id = 6 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1G "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 7, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 7 AND o_w_id = :w_id AND o_d_id = 7 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1H "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 8, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 8 AND o_w_id = :w_id AND o_d_id = 8 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1I "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 9, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 9 AND o_w_id = :w_id AND o_d_id = 9 AND \
   o_id = no_o_id AND rownum <= 1 UNION ALL \
"

#define SQLTXT1J "\
SELECT /*+ USE_NL(NEW_ORDER ORDERS) ORDERED */ 10, no_o_id, new_order.rowid, o_c_id,
orders.rowid \
FROM new_order, orders \
WHERE no_w_id = :w_id AND no_d_id = 10 AND o_w_id = :w_id AND o_d_id = 10 AND \
   o_id = no_o_id AND rownum <= 1"

#define SQLTXT2 "DELETE FROM new_order WHERE rowid = :no_rowid"
#endif

#ifdef DMLRETDEL
#define SQLTXT3 "UPDATE orders SET o_carrier_id = :carrier_id \
                 WHERE o_id = :o_id and o_d_id = :d_id and o_w_id = :w_id \
                 returning o_c_id into :o_c_id"
#else
#define SQLTXT3 "UPDATE orders SET o_carrier_id = :carrier_id \
   WHERE rowid = :o_rowid"
#endif

#ifdef DMLRETDEL
#define SQLTXT4 "UPDATE /*+ buffer */ order_line SET ol_delivery_d = :cr_date \
   WHERE ol_w_id = :w_id AND ol_d_id = :d_id AND ol_o_id = :o_id \
   RETURNING ol_amount into :ol_amount "
#else
#define SQLTXT4 "UPDATE order_line SET ol_delivery_d = :cr_date \
   WHERE ol_w_id = :w_id AND ol_d_id = :d_id AND ol_o_id = :o_id"

#define SQLTXT5A "\
SELECT :d_id1, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id1 AND ol_o_id = :o_id1 UNION ALL \
SELECT :d_id2, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id2 AND ol_o_id = :o_id2 UNION ALL \
"
```

```
#define SQLTXT5B "\
SELECT :d_id3, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id3 AND ol_o_id = :o_id3 UNION ALL \
SELECT :d_id4, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id4 AND ol_o_id = :o_id4 UNION ALL \
"

#define SQLTXT5C "\
SELECT :d_id5, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id5 AND ol_o_id = :o_id5 UNION ALL \
SELECT :d_id6, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id6 AND ol_o_id = :o_id6 UNION ALL \
"

#define SQLTXT5D "\
SELECT :d_id7, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id7 AND ol_o_id = :o_id7 UNION ALL \
SELECT :d_id8, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id8 AND ol_o_id = :o_id8 UNION ALL \
"

#define SQLTXT5E "\
SELECT :d_id9, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id9 AND ol_o_id = :o_id9 UNION ALL \
SELECT :d_id10, SUM(ol_amount) FROM order_line WHERE ol_w_id = :w_id AND \
 ol_d_id = :d_id10 AND ol_o_id = :o_id10"

#endif
#endif /* PLSQLDEL */

#define SQLTXT6 "UPDATE customer SET c_balance = c_balance + :amt, \
   c_delivery_cnt = c_delivery_cnt + 1 WHERE c_w_id = :w_id AND \
   c_d_id = :d_id AND c_id = :c_id"

#define NDISTS 10
#define ROWIDLEN 20


struct delctx {
   sb2 del_o_id_ind[NDISTS];
   sb2 cons_ind[NDISTS];
   sb2 w_id_ind[NDISTS];
   sb2 d_id_ind[NDISTS];
   sb2 c_id_ind[NDISTS];
   sb2 del_date_ind[NDISTS];
   sb2 carrier_id_ind[NDISTS];
   sb2 amt_ind[NDISTS];
   sb2 no_rowid_ind[NDISTS];
   sb2 o_rowid_ind[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   sb2 inum_ind;
#endif

#ifdef DMLRETDEL
   ub4 del_o_id_len[NDISTS];
   ub4 c_id_len[NDISTS];
   int oid_ctx;
   int cid_ctx;
   OCIBind *olamt_bp;
#else
   ub2 del_o_id_len[NDISTS];
   ub2 c_id_len[NDISTS];
#endif

   ub2 cons_len[NDISTS];
   ub2 w_id_len[NDISTS];
   ub2 d_id_len[NDISTS];
   ub2 del_date_len[NDISTS];
   ub2 carrier_id_len[NDISTS];
   ub2 amt_len[NDISTS];
   ub2 no_rowid_len[NDISTS];
   ub2 no_rowid_ptr_len[NDISTS];
   ub2 o_rowid_len[NDISTS];
   ub2 o_rowid_ptr_len[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   ub2 inum_len;
#endif

   ub2 del_o_id_rcode[NDISTS];
   ub2 cons_rcode[NDISTS];
   ub2 w_id_rcode[NDISTS];
   ub2 d_id_rcode[NDISTS];
   ub2 c_id_rcode[NDISTS];
   ub2 del_date_rcode[NDISTS];
   ub2 carrier_id_rcode[NDISTS];
   ub2 amt_rcode[NDISTS];
   ub2 no_rowid_rcode[NDISTS];
   ub2 o_rowid_rcode[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   ub2 inum_rcode;
#endif

   int del_o_id[NDISTS];
   int cons[NDISTS];
   int w_id[NDISTS];
   int d_id[NDISTS];
   int c_id[NDISTS];
   int carrier_id[NDISTS];
   int amt[NDISTS];
   ub4 del_o_id_rcnt;
   int retry;
   OCIRowid *no_rowid_ptr[NDISTS];
   OCIRowid *o_rowid_ptr[NDISTS];
   OCIDate del_date[NDISTS];
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   char inum[10];
#endif
   OCIStmt *curd0;
   OCIStmt *curd1;
   OCIStmt *curd2;
   OCIStmt *curd3;
   OCIStmt *curd4;
   OCIStmt *curd5;
   OCIStmt *curd6;
   OCIStmt *curdtest;

   OCIBind *w_id_bp;
   OCIBind *w_id_bp3;
   OCIBind *w_id_bp4;
   OCIBind *w_id_bp5;
   OCIBind *w_id_bp6;
   OCIBind *d_id_bp;
   OCIBind *d_id_bp3;
   OCIBind *d_id_bp4;
   OCIBind *d_id_bp6;
   OCIBind *o_id_bp;
   OCIBind *cr_date_bp;
   OCIBind *c_id_bp;
   OCIBind *c_id_bp3;
   OCIBind *no_rowid_bp;
   OCIBind *carrier_id_bp;
   OCIBind *o_rowid_bp;
   OCIBind *del_o_id_bp;
   OCIBind *del_o_id_bp3;
   OCIBind *amt_bp;
   OCIBind *bstr1_bp[10];
   OCIBind *bstr2_bp[10];
   OCIBind *retry_bp;
   OCIDefine *inum_dp;
   OCIDefine *d_id_dp;
   OCIDefine *del_o_id_dp;
   OCIDefine *no_rowid_dp;
   OCIDefine *c_id_dp;
   OCIDefine *o_rowid_dp;
   OCIDefine *cons_dp;
   OCIDefine *amt_dp;

   int norow;
};

typedef struct delctx delctx;

delctx *dctx;

#ifdef DMLRETDEL
struct amtctx {
  int ol_amt[NDISTS][NITEMS];
  sb2 ol_amt_ind[NDISTS][NITEMS];
```

```
  ub4 ol_amt_len[NDISTS][NITEMS];
  ub2 ol_amt_rcode[NDISTS][NITEMS];
  int ol_cnt[NDISTS];
};
typedef struct amtctx amtctx;
amtctx *actx;

#endif


#ifdef DMLRETDEL
extern sb4 no_data();

sb4 TPC_oid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  *bufpp = &dctx->del_o_id[iter];
  *indpp= &dctx->del_o_id_ind[iter];
  dctx->del_o_id_len[iter]=sizeof(dctx->del_o_id[0]);
  *alenp= &dctx->del_o_id_len[iter];
  *rcodepp = &dctx->del_o_id_rcode[iter];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}
sb4 cid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  *bufpp = &dctx->c_id[iter];
  *indpp= &dctx->c_id_ind[iter];
  dctx->c_id_len[iter]=sizeof(dctx->c_id[0]);
  *alenp= &dctx->c_id_len[iter];
  *rcodepp = &dctx->c_id_rcode[iter];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}


sb4 amt_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
               dvoid **bufpp, ub4 **alenp, ub1 *piecep,
               dvoid **indpp, ub2 **rcodepp)
{
  amtctx *actx;
  actx =(amtctx*)ctxp;
  actx->ol_cnt[iter]=actx->ol_cnt[iter]+1;
  *bufpp = &actx->ol_amt[iter][index];
  *indpp= &actx->ol_amt_ind[iter][index];
  actx->ol_amt_len[iter][index]=sizeof(actx->ol_amt[0][0]);
  *alenp= &actx->ol_amt_len[iter][index];
  *rcodepp = &actx->ol_amt_rcode[iter][index];
  *piecep =OCI_ONE_PIECE;
  return (OCI_CONTINUE);
}

#endif


tkvcdinit ()

{

   int i,j;
   char bstr1[10];
   char bstr2[10];
   text stmbuf[SQL_BUF_SIZE];

   dctx = (delctx *) malloc (sizeof(delctx));
   memset(dctx,(char)0,sizeof(delctx));
   dctx->norow = 0;
#ifdef DMLRETDEL
   actx = (amtctx *) malloc (sizeof(amtctx));
   memset(actx,(char)0,sizeof(amtctx));
#else
   for(i=0;i<NDISTS;i++) {
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&dctx->o_rowid_ptr[i],
              OCI_DTYPE_ROWID,0,(dvoid**)0));
    OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,(dvoid**)&dctx->no_rowid_ptr[i],
              OCI_DTYPE_ROWID,0,(dvoid**)0));
   }
#endif

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd0), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT0);
   OCIStmtPrepare(dctx->curd0, errhp, stmbuf, strlen((char *)stmbuf),OCI_NTV_SYNTAX,
OCI_DEFAULT);

   OCIDFNRA(dctx->curd0, dctx->inum_dp,errhp,1,dctx->inum,SIZ(dctx->inum),SQLT_STR,
          &(dctx->inum_ind),&(dctx->inum_len),&(dctx->inum_rcode));
#endif

#ifdef PLSQLDEL
   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd0), OCI_HTYPE_STMT,
                  0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT);
   OCIStmtPrepare(dctx->curd0, errhp, stmbuf, strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX, OCI_DEFAULT);
   OCIBND(dctx->curd0, dctx->w_id_bp , errhp,":w_id",ADR(w_id),SIZ(int),
          SQLT_INT);
   OCIBND(dctx->curd0, dctx->carrier_id_bp , errhp,":carrier_id",
          ADR(dctx->carrier_id), SIZ(int), SQLT_INT);

   OCIBNDRAA(dctx->curd0, dctx->o_id_bp, errhp, ":order_id",
            dctx->del_o_id,SIZ(int),SQLT_INT, dctx->del_o_id_ind,
            dctx->del_o_id_len,dctx->del_o_id_rcode,NDISTS,
            &dctx->del_o_id_rcnt);
   OCIBND(dctx->curd0, dctx->retry_bp , errhp,":retry",ADR(dctx->retry),
          SIZ(int),SQLT_INT);
#else
# ifdef DMLRETDEL
   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd1), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, "%s",    SQLTXT1);
   OCIStmtPrepare(dctx->curd1, errhp, stmbuf, strlen((char *)stmbuf),OCI_NTV_SYNTAX,
OCI_DEFAULT);

   OCIBND(dctx->curd1, dctx->w_id_bp,errhp,":w_id",dctx->w_id,SIZ(int),
          SQLT_INT);
   OCIBNDRA(dctx->curd1, dctx->d_id_bp,errhp,":d_id",dctx->d_id,SIZ(int),
          SQLT_INT,NULL,NULL,NULL);

   OCIBNDRAD(dctx->curd1, dctx->del_o_id_bp, errhp, ":o_id",
             SIZ(int),SQLT_INT,NULL,
             &dctx->oid_ctx,no_data,TPC_oid_data);
# else

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd1), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, "%s%s%s%s%s%s%s%s%s%s",    SQLTXT1A,
                                                                SQLTXT1B,
                                                                SQLTXT1C,
                                                                SQLTXT1D,
                                                                SQLTXT1E,
                                                                SQLTXT1F,
                                                                SQLTXT1G,
                                                                SQLTXT1H,
                                                                SQLTXT1I,
                                                                SQLTXT1J
          );
   OCIStmtPrepare(dctx->curd1, errhp, stmbuf, strlen((char *)stmbuf),OCI_NTV_SYNTAX,
OCI_DEFAULT);

   OCIERROR(errhp,
      OCIAttrSet(dctx->curd1,OCI_HTYPE_STMT,(dvoid*)&dctx->norow,0,
                 OCI_ATTR_PREFETCH_ROWS,errhp));



   /* bind variables */

   OCIBND(dctx->curd1, dctx->w_id_bp,errhp,":w_id",ADR(w_id),SIZ(int),SQLT_INT);

   OCIDFNRA(dctx->curd1, dctx->d_id_dp,errhp,1,dctx->d_id,SIZ(int),
            SQLT_INT, dctx->d_id_ind,dctx->d_id_len,dctx->d_id_rcode);
   OCIDFNRA(dctx->curd1, dctx->del_o_id_dp,errhp,2,dctx->del_o_id,
            SIZ(int), SQLT_INT,dctx->del_o_id_ind,
            dctx->del_o_id_len, dctx->del_o_id_rcode);
   OCIDFNRA(dctx->curd1, dctx->no_rowid_dp,errhp,3,dctx->no_rowid_ptr,
            SIZ(OCIRowid *), SQLT_RDD,dctx->no_rowid_ind,
```

```
            dctx->no_rowid_len, dctx->no_rowid_rcode);
   OCIDFNRA(dctx->curd1, dctx->c_id_dp,errhp,4,dctx->c_id,SIZ(dctx->c_id[0]),
            SQLT_INT, dctx->c_id_ind,dctx->c_id_len,dctx->c_id_rcode);
   OCIDFNRA(dctx->curd1, dctx->o_rowid_dp,errhp,5,dctx->o_rowid_ptr,
            SIZ(OCIRowid *), SQLT_RDD,dctx->o_rowid_ind,
            dctx->o_rowid_len, dctx->o_rowid_rcode);

   /* open second cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd2), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT2);
   OCIStmtPrepare(dctx->curd2, errhp, stmbuf, strlen((char *)stmbuf),
                     OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */
   OCIBNDRA(dctx->curd2, dctx->no_rowid_bp,errhp,":no_rowid",&(dctx-
>no_rowid_ptr[0]),
            SIZ(dctx->no_rowid_ptr[0]),SQLT_RDD,dctx->no_rowid_ind,
            dctx->no_rowid_len,dctx->no_rowid_rcode);

# endif /*DMLRETDEL*/

   /* open third cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd3), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT3);
   OCIStmtPrepare(dctx->curd3, errhp, stmbuf, strlen((char *)stmbuf),
                     OCI_NTV_SYNTAX, OCI_DEFAULT);


   /* bind variables */

   OCIBNDRA(dctx->curd3, dctx->carrier_id_bp,errhp,":carrier_id",dctx->carrier_id,
          SIZ(dctx->carrier_id[0]),SQLT_INT,dctx->carrier_id_ind,
          dctx->carrier_id_len,dctx->carrier_id_rcode);

# ifdef DMLRETDEL
   OCIBNDRA(dctx->curd3, dctx->w_id_bp3, errhp, ":w_id", dctx->w_id,SIZ(int),
            SQLT_INT, NULL, NULL, NULL);
   OCIBNDRA(dctx->curd3, dctx->d_id_bp3, errhp, ":d_id", dctx->d_id,SIZ(int),
            SQLT_INT,NULL, NULL, NULL);
   OCIBNDRA(dctx->curd3, dctx->del_o_id_bp3, errhp, ":o_id", dctx->del_o_id,
            SIZ(int), SQLT_INT,NULL,NULL,NULL);
   OCIBNDRAD(dctx->curd3, dctx->c_id_bp3, errhp, ":o_c_id", SIZ(int),
            SQLT_INT,NULL,&dctx->cid_ctx,no_data, cid_data);
# else

   OCIBNDRA(dctx->curd3, dctx->o_rowid_bp,errhp,":o_rowid",&(dctx->o_rowid_ptr[0]),
              SIZ(dctx->o_rowid_ptr[0]),SQLT_RDD,dctx->o_rowid_ind,
              dctx->o_rowid_ptr_len,dctx->o_rowid_rcode);
#endif

   /* open fourth cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd4), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT4);
   OCIStmtPrepare(dctx->curd4, errhp, stmbuf, strlen((char *)stmbuf),
                     OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */

   OCIBND(dctx->curd4, dctx->w_id_bp4,errhp,":w_id",dctx->w_id,
          SIZ(int), SQLT_INT);
   OCIBND(dctx->curd4, dctx->d_id_bp4,errhp,":d_id",dctx->d_id,
          SIZ(int), SQLT_INT);
   OCIBND(dctx->curd4, dctx->o_id_bp,errhp,":o_id",dctx->del_o_id,
            SIZ(int),SQLT_INT);
   OCIBND(dctx->curd4, dctx->cr_date_bp,errhp,":cr_date", dctx->del_date,
            SIZ(OCIDate), SQLT_ODT);
# ifdef DMLRETDEL
   OCIBNDRAD(dctx->curd4, dctx->olamt_bp, errhp, ":ol_amount",
             SIZ(int), SQLT_INT,NULL, actx,no_data,amt_data);
# else



   /* open fifth cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd5), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, "%s%s%s%s%s",        SQLTXT5A,
                                                            SQLTXT5B,
                                                            SQLTXT5C,
                                                            SQLTXT5D,
                                                            SQLTXT5E
   );
   OCIStmtPrepare(dctx->curd5, errhp, stmbuf, strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX, OCI_DEFAULT);

   OCIERROR(errhp,
      OCIAttrSet(dctx->curd5,OCI_HTYPE_STMT,(dvoid*)&dctx->norow,0,
                  OCI_ATTR_PREFETCH_ROWS,errhp));

   /* bind variables */

   OCIBND(dctx->curd5,dctx->w_id_bp,errhp,":w_id",ADR(w_id),SIZ(w_id),SQLT_INT);
   for (i = 0; i < NDISTS; i++) {
      sprintf (bstr1, ":d_id%d", i + 1);
      sprintf (bstr2, ":o_id%d", i + 1);
      OCIBNDRA(dctx->curd5,dctx->bstr1_bp[i],errhp,bstr1,ADR(dctx->d_id[i]),
                  SIZ(dctx->d_id[0]),SQLT_INT, &(dctx->d_id_ind[i]),
                  &(dctx->d_id_len[i]),&(dctx->d_id_rcode[i]));
      OCIBNDRA(dctx->curd5,dctx->bstr2_bp[i],errhp,bstr2,ADR(dctx->del_o_id[i]),
                  SIZ(dctx->del_o_id[0]),SQLT_INT, &(dctx->del_o_id_ind[i]),
                  &(dctx->del_o_id_len[i]),&(dctx->del_o_id_rcode[i]));
   }

   OCIDFNRA(dctx->curd5,dctx->cons_dp,errhp,1,dctx->cons,SIZ(dctx-
>cons[0]),SQLT_INT,
          dctx->cons_ind,dctx->cons_len,dctx->cons_rcode);
   OCIDFNRA(dctx->curd5,dctx->amt_dp,errhp,2,dctx->amt,SIZ(dctx->amt[0]),SQLT_INT,
          dctx->amt_ind,dctx->amt_len,dctx->amt_rcode);
#endif
   /* open sixth cursor */

   OCIHandleAlloc(tpcenv, (dvoid **)(&dctx->curd6), OCI_HTYPE_STMT, 0, (dvoid**)0);
   sprintf ((char *) stmbuf, SQLTXT6);
   OCIStmtPrepare(dctx->curd6, errhp, stmbuf, strlen((char *)stmbuf),
                        OCI_NTV_SYNTAX, OCI_DEFAULT);

   /* bind variables */


   OCIBND(dctx->curd6,dctx->amt_bp,errhp,":amt",dctx->amt,SIZ(int),
          SQLT_INT);
   OCIBND(dctx->curd6,dctx->w_id_bp6,errhp,":w_id",dctx->w_id,SIZ(int),
            SQLT_INT);
   OCIBND(dctx->curd6,dctx->d_id_bp6,errhp,":d_id",dctx->d_id,SIZ(int),
            SQLT_INT);
   OCIBND(dctx->curd6,dctx->c_id_bp,errhp,":c_id",dctx->c_id,SIZ(int),
             SQLT_INT);
#endif
   return (0);

}


void shiftdata(from)
int from ;
{
   int i;
   for (i=from;i<NDISTS-1; i++)
   {
     dctx->del_o_id_ind[i] = dctx->del_o_id_ind[i+1];
     dctx->del_o_id[i] = dctx->del_o_id[i+1];
     dctx->w_id[i] = dctx->w_id[i+1];
     dctx->d_id[i] = dctx->d_id[i+1];
     dctx->carrier_id[i] = dctx->carrier_id[i+1];
   }
}


tkvcd ()

{

   int i, j,v;
   int rpc,rcount,count;
   int invalid;
```

```
   int tmp_id;
   int tmp_amt;

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   int hasno;
   int reread;
   char sdate[30];

   OCIStmtExecute(tpcsvc,dctx->curd0,errhp,1,0,0,0,OCI_DEFAULT);
   sysdate (sdate);
   printf ("Delivery started at %s on node %s\n", sdate, dctx->inum);
#endif
#ifdef PLSQLDEL
   for (i = 0; i < NDISTS; i++)
   {
      dctx->del_o_id_ind[i] = TRUE;
      dctx->del_o_id_len[i] = sizeof(int);
   }

   OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,dctx->curd0,errhp,1,0,0,0,OCI_DEFAULT));

   for (i = 0; i < NDISTS; i++)
   {
      del_o_id[i] = 0;
      if (dctx->del_o_id_ind[i] == 0)
      {
        del_o_id[i] = dctx->del_o_id[i];
      }
   }
#else

retry:

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   reread = 1;
#endif

iso:

   invalid = 0;

   /* initialization for array operations */

   for (i = 0; i < NDISTS; i++) {
      dctx->del_o_id_ind[i] = TRUE;
      dctx->cons_ind[i] = TRUE;
      dctx->w_id_ind[i] = TRUE;
      dctx->d_id_ind[i] = TRUE;
      dctx->c_id_ind[i] = TRUE;
      dctx->del_date_ind[i] = TRUE;
      dctx->carrier_id_ind[i] = TRUE;
      dctx->amt_ind[i] = TRUE;
      dctx->no_rowid_ind[i] = TRUE;
      dctx->o_rowid_ind[i] = TRUE;

      dctx->del_o_id_len[i] = SIZ(dctx->del_o_id[0]);
      dctx->cons_len[i] = SIZ(dctx->cons[0]);
      dctx->w_id_len[i] = SIZ(dctx->w_id[0]);
      dctx->d_id_len[i] = SIZ(dctx->d_id[0]);
      dctx->c_id_len[i] = SIZ(dctx->c_id[0]);
      dctx->del_date_len[i] = DEL_DATE_LEN;
      dctx->carrier_id_len[i] = SIZ(dctx->carrier_id[0]);
      dctx->amt_len[i] = SIZ(dctx->amt[0]);
      dctx->no_rowid_len[i] = ROWIDLEN;
      dctx->o_rowid_len[i] = ROWIDLEN;
      dctx->o_rowid_ptr_len[i] = SIZ(dctx->o_rowid_ptr[0]);
      dctx->no_rowid_ptr_len[i] = SIZ(dctx->no_rowid_ptr[0]);

      dctx->w_id[i] = w_id;
      dctx->d_id[i] = i+1;
      dctx->carrier_id[i] = o_carrier_id;
      memcpy(&dctx->del_date[i],&cr_date,sizeof(OCIDate));
   }

#ifdef DMLRETDEL
   memset(actx,(char)0,sizeof(amtctx));
#endif /* DMLRETDEL */
   /* array select from new_order and orders tables */
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd1,errhp,NDISTS,0,0,0,OCI_DEFAULT);
   if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA)) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }
   /* mark districts with no new order */
   OCIAttrGet(dctx->curd1,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);
   rpc = rcount;
#ifdef DMLRETDEL   /* we have to compress the array here */
   if (rcount != NDISTS )
   {
     int j = 0;
     for (i=0;i < NDISTS; i++)
     {
      if (dctx->del_o_id_ind[j] == 0) /* there is data here */
        j++;
      else
        shiftdata(j);
     }
   }
#else
   invalid = NDISTS - rcount;
   for (i = rpc; i < NDISTS; i++) {
      dctx->del_o_id_ind[i] = NA;
      dctx->w_id_ind[i] = NA;
      dctx->d_id_ind[i] = NA;
      dctx->c_id_ind[i] = NA;
      dctx->carrier_id_ind[i] = NA;
      dctx->no_rowid_ind[i] = NA;
      dctx->o_rowid_ind[i] = NA;
   }
#endif

#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
   if (invalid) {
      sysdate (sdate);
      for (i = 1; i <= NDISTS; i++) {
         hasno = 0;
         for (j = 0; j < rpc; j++) {
            if (dctx->d_id[j] == i) {
               hasno = 1;
               break;
            }
         }
         if (!hasno)
            printf ("Delivery [dist %d] found no new order at %s\n", i, sdate);
      }
      if (reread) {
         sleep (60);
         sysdate (sdate);
         printf ("Delivery wake up at %s\n", sdate);
         reread = 0;
         goto iso;
      }
   }
#endif

#ifndef DMLRETDEL
   /* array delete of new_order table */
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd2,errhp,rpc,0,0,0,OCI_DEFAULT);
   if(execstatus != OCI_SUCCESS) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
```

```
      } else {
         return -1;
      }
   }

   /* mark districts with no new order */
   OCIAttrGet(dctx->curd2,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);

   if (rcount != rpc) {
      message ("Error in TPC-C server %d: %d rows selected, %d rows deleted\n",
               proc_no, rpc, dctx->curd2.rpc);
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (DEL_ERROR);
   }
#endif /* DMLRETDEL */

   execstatus=OCIStmtExecute(tpcsvc,dctx->curd3,errhp,rpc,0,0,0,OCI_DEFAULT);
   if(execstatus != OCI_SUCCESS) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }

   OCIAttrGet(dctx->curd3,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);

   if (rcount != rpc) {
      message ("Error in TPC-C server %d: %d rows selected, %d ords updated\n",
               proc_no, rpc, rcount);
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (-1);
   }

   /* array update of order_line table */
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd4,errhp,rpc,0,0,0,OCI_DEFAULT);
   if(execstatus != OCI_SUCCESS) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }
#ifdef DMLRETDEL
  OCIAttrGet(dctx->curd4,OCI_HTYPE_STMT,&rcount,NULL,OCI_ATTR_ROW_COUNT,errhp);
/* add up amounts */
   count=0;
   for (i=0;i<rpc;i++)
   {
     dctx->amt[i]=0;
     for (j=0;j<actx->ol_cnt[i];j++)
       if ( actx->ol_amt_rcode[i][j] == 0)
       {
         dctx->amt[i] = dctx->amt[i] + actx->ol_amt[i][j];
         count = count+1;
       }
   }
   if (rcount > rpc*NITEMS) {
      message ("Error in TPC-C server %d: %d ordnrs updated, %d ordl updated\n",
               proc_no, rpc, rcount);
   }
#else
   /* array select from order_line table */
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd5,errhp,rpc,0,0,0,OCI_DEFAULT);
   if((execstatus != OCI_SUCCESS) && (execstatus != OCI_NO_DATA)) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }

   OCIAttrGet(dctx->curd5,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);
   if (rcount != rpc) {
      message ("Error in TPC-C server %d: %d rows selected, %d ordl selected\n",
               proc_no, rpc, rcount);
      OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
      return (-1);
   }

   /* reorder amount selected if necessary */

   for (i = 0; i < rpc; i++) {
      if (dctx->cons[i] != dctx->d_id[i]) {
         message ("TPC-C server %d: reordering amount\n", proc_no);
         for (j = i + 1; j < rpc; j++) {
            if (dctx->cons[j] == dctx->d_id[i]) {
               tmp_id = dctx->cons[i];
               dctx->cons[i] = dctx->cons[j];
               dctx->cons[j] = tmp_id;
               tmp_amt = dctx->amt[i];
               dctx->amt[i] = dctx->amt[j];
               dctx->amt[j] = tmp_amt;
               break;
            }
         }
         if (j >= rpc) {
            message ("Error in TPC-C server %d: missing ordl?\n", proc_no);
              OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
            return (-1);
         }
      }
   }
#endif
#if defined(ISO5) || defined(ISO6)
   printf ("d_id:amount\n");
   for (i = 0; i < rpc; i++)
      printf ("%d:%.2f ", dctx->d_id[i], (float)dctx->amt[i]/100);
   printf ("\n");
#endif

   /* array update of customer table */
#if defined(ISO5) || defined (ISO6)
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd6,errhp,rpc,0,0,0,
                                          OCI_DEFAULT);
#else
   execstatus=OCIStmtExecute(tpcsvc,dctx->curd6,errhp,rpc,0,0,0,
                                    OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);
#endif

   if(execstatus != OCI_SUCCESS) {
     OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
     errcode = OCIERROR(errhp,execstatus);
     if(errcode == NOT_SERIALIZABLE) {
        retries++;
        goto retry;
     } else if (errcode == RECOVERR) {
        retries++;
        goto retry;
     } else {
        return -1;
     }
   }

   OCIAttrGet(dctx->curd6,OCI_HTYPE_STMT,&rcount,0,OCI_ATTR_ROW_COUNT,errhp);

   if (rcount != rpc) {
      message ("Error in TPC-C server %d: %d rows selected, %d cust updated\n",
               proc_no, rpc, rcount);
     OCITransRollback(tpcsvc, errhp, OCI_DEFAULT);
```

```
      return (-1);
   }

#if defined(ISO5) || defined(ISO6)
   sysdate (sdate);
#ifdef ISO5
   printf ("Delivery sleep before commit at %s\n", sdate);
#else
   printf ("Delivery sleep before abort at %s\n", sdate);
#endif
   sleep (60);
   sysdate (sdate);
   printf ("Delivery wake up at %s\n", sdate);
#endif

#ifdef ISO6
   printf("Delivery ISO6 Rolling back.\n");
   OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
#endif

#ifdef ISO5
   OCITransCommit(tpcsvc, errhp, OCI_DEFAULT);
#endif

#if defined(ISO5) || defined(ISO6)
   sysdate (sdate);
   printf ("Delivery completed at: %s\n", sdate);
#endif

   /* return o_id's in district id order */

   for (i = 0; i < NDISTS; i++)
      del_o_id[i] = 0;
   for (i = 0; i < rpc; i++)
      del_o_id[dctx->d_id[i] - 1] = dctx->del_o_id[i];
#endif

   return (0);

}



void tkvcddone ()

{

   if (dctx)
   {
#if defined(ISO) || defined(ISO5) || defined(ISO6) || defined(ISO8)
     OCIHandleFree((dvoid *)dctx->curd0,OCI_HTYPE_STMT);
#endif
#ifdef PLSQLDEL
     OCIHandleFree((dvoid *)dctx->curd0,OCI_HTYPE_STMT);
#else
     OCIHandleFree((dvoid *)dctx->curd1,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd2,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd3,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd4,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd5,OCI_HTYPE_STMT);
     OCIHandleFree((dvoid *)dctx->curd6,OCI_HTYPE_STMT);
#endif
     free (dctx);
   }

}
```

client/oracle/ora_tpcc.h

```
/*
 * $Header: tpcc.h 7030100.1 95/07/19 15:10:55 plai Generic<base> $ Copyr (c) 1993
Oracle
 */
/*+=================================================================+
 |          Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
 |                       OPEN SYSTEMS PERFORMANCE GROUP              |
 |                          All Rights Reserved                      |
 +=================================================================+
 | FILENAME
 |    tpcc.h
 | DESCRIPTION
 |    Include file for TPC-C benchmark programs.
 +=================================================================*/

#ifndef TPCC_H
#define TPCC_H


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

#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>

/***************/
#define DMLRETDEL
#define PLSQLNO
/***************/

typedef struct cda_def csrdef;
typedef struct cda_def ldadef;


/* TPC-C transaction functions */

extern int TPCinit ();
extern int TPCnew ();
extern int TPCpay ();
extern int TPCord ();
extern int TPCdel ();
extern int TPCsto ();
extern int TPCexit ();
extern int TPCdumpinit ();
extern int TPCdumpnew ();
extern int TPCdumppay ();
extern int TPCdumpord ();
extern int TPCdumpdel ();
extern int TPCdumpsto ();
extern int TPCdumpexit ();

#define OCI_HTYPE_STM           OCI_HTYPE_STMT
#define OCI_ATTR_ROWCNT         OCI_ATTR_ROW_COUNT
#define OCI_HTYPE_ERR           OCI_HTYPE_ERROR
#define OCI_ATTR_SRVRCTXT       OCI_ATTR_SERVER
#define OCI_ATTR_USERCTXT       OCI_ATTR_SESSION


/* Error codes */

#define RECOVERR -10
#define IRRECERR -20
#define NOERR    111
#define DEL_ERROR -666
#define DEL_DATE_LEN 7
#define NDISTS 10
#define NITEMS 15
#define SQL_BUF_SIZE 8192

#define FULLDATE "dd-mon-yy.hh:mi:ss"
#define SHORTDATE "dd-mm-yyyy"



#define DELRT 80.0

extern int tkvcninit ();
extern int tkvcpnit ();
extern int tkvcoinit ();
extern int tkvcdinit ();
extern int tkvcsinit ();
```

```
extern int tkvcn ();
extern int tkvcp ();
extern int tkvco ();
extern int tkvcd ();
extern int tkvcs ();

extern void tkvcndone ();
extern void tkvcpdone ();
extern void tkvcodone ();
extern void tkvcddone ();
extern void tkvcsdone ();

extern int tkvcss (); /* for alter session to get memory size and trace */
extern boolean multitranx;
extern int ord_init;


extern errrpt ();
extern int ocierror(char *fname, int lineno,OCIError *errhp, sword status);
extern int sqlfile(char *fname, text *linebuf);

extern FILE *lfp;
extern FILE *fopen ();
extern int proc_no;
extern int doid[];

extern int execstatus;
extern int errcode;

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCIStmt *curntest;
/* The bind and define handles for each transaction are
   included in their respective header files. */



/* for stock-level transaction */

extern int w_id;
extern int d_id;
extern int c_id;
extern int threshold;
extern int low_stock;

/* for delivery transaction */

extern int del_o_id[10];
extern int carrier_id;
extern int retries;

/* for order-status transaction */

extern int bylastname;
extern char c_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern text o_entry_d[20];
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
extern int ol_quantity[15];
extern int ol_amount[15];
ub4    ol_del_len[15];
extern text ol_delivery_d[15][11];

/* for payment transaction */

extern int c_w_id;
extern int c_d_id;
extern int h_amount;
extern char w_street_1[21];
extern char w_street_2[21];
extern char w_city[21];
extern char w_state[3];
extern char w_zip[10];
extern char d_street_1[21];
extern char d_street_2[21];
extern char d_city[21];
extern char d_state[3];
extern char d_zip[10];
extern char c_street_1[21];
extern char c_street_2[21];
extern char c_city[21];
extern char c_state[3];
extern char c_zip[10];
extern char c_phone[17];
extern text c_since_d[11];
extern char c_credit[3];
extern int c_credit_lim;
extern float c_discount;
extern char c_data[201];
extern text h_date[20];

/* for new order transaction */

extern int nol_i_id[15];
extern int nol_supply_w_id[15];
extern int nol_quantity[15];
extern int nol_quanti10[15];
extern int nol_quanti91[15];
extern int nol_ytdqty[15];
extern int nol_amount[15];
extern int o_all_local;
extern float w_tax;
extern float d_tax;
extern float total_amount;
extern char i_name[15][25];
extern int i_name_strlen[15];
extern ub2 i_name_strlen_len[15];
extern ub2 i_name_strlen_rcode[15];
extern ub4 i_name_strlen_csize;
extern int s_quantity[15];
extern char brand_gen[15];
extern ub2 brand_gen_len[15];
extern ub2 brand_gen_rcode[15];
extern ub4 brand_gen_csize;
extern int i_price[15];
extern char brand_generic[15][1];
extern int status;

/* Miscellaneous */
extern OCIDate cr_date;
extern OCIDate c_since;
extern OCIDate o_entry_d_base;
extern OCIDate ol_d_base[15];

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#define VER7            2

#define NA              -1      /* ANSI SQL NULL */
#define NLT             1       /* length for string null terminator */
#define DEADLOCK        60      /* ORA-00060: deadlock */
#define NO_DATA_FOUND   1403    /* ORA-01403: no data found */
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not serializable */
#define SNAPSHOT_TOO_OLD  1555  /* ORA-01555: snapshot too old */

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))

typedef char date[24+NLT];
```

```
typedef char varchar2;

#define min(x,y)  (((x) < (y)) ? (x) : (y))

#define OCIERROR(errp,function)\
        ocierror(__FILE__,__LINE__,(errp),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, progv, progvl, ftype)\
       ocierror(__FILE__,__LINE__,(errp), \
             OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__, (errp), \
             OCIBindByName((stmp), &(bndp), (errp), \
                         (text *)(sqlvar), strlen((sqlvar)),\
                         (progv), (progvl), (ftype),0,0,0,0,0,OCI_DEFAULT));

#define OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindByName((stmp),&(bndp),(errp),(text *)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

#define OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,cbf_nodata,cbf_data)
\
       ocierror(__FILE__,__LINE__,(errp), \
            OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindByName((stmp),&(bndp),(errp),(text *)(sqlvar), \
                          strlen((sqlvar)),0,(progvl),(ftype), \
                          indp,0,0,0,0,OCI_DATA_AT_EXEC)); \
       ocierror(__FILE__,__LINE__,(errp), \
            OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodata),(ctxp),(cbf_data)));



#define OCIBNDR(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
       ocierror(__FILE__,__LINE__,(errp), \
          OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
       ocierror(__FILE__,__LINE__,(errp), \
          OCIBindByName((stmp),&(bndp),(errp),(text *)(sqlvar),strlen((sqlvar)),\

(progv),(progvl),(ftype),(indp),(alen),(arcode),0,0,OCI_DEFAULT));

#define OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode,ms,cu) \
       ocierror(__FILE__,__LINE__, (errp), \
         OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0));\
       ocierror(__FILE__,__LINE__,(errp),\
         OCIBindByName((stmp),&(bndp),(errp),(text *)(sqlvar),strlen((sqlvar)),\
       (progv),(progvl),(ftype),(indp),(alen),(arcode),(ms),(cu),OCI_DEFAULT));

#define OCIDEFINE(stmp,dfnp,errp,pos,progv,progvl,ftype)\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),(ftype),\
                      0,0,0,OCI_DEFAULT);


#define OCIDEF(stmp,dfnp,errp,pos,progv,progvl,ftype) \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                          (dvoid**)0);\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),\
                        (ftype),NULL,NULL,NULL,OCI_DEFAULT); \


#define OCIDFNRA(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,alen,arcode) \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                          (dvoid**)0);\
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),\
                                          (progvl),(ftype),(indp),(alen),\
                                          (arcode),OCI_DEFAULT);

#define OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,ctxp,cbf_data) \
       ocierror(__FILE__,__LINE__,(errp), \
       OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                          (dvoid**)0));\
       ocierror(__FILE__,__LINE__,(errp), \
       OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv), (progvl),(ftype),\
                                    (indp),NULL,NULL, OCI_DYNAMIC_FETCH));\
       ocierror(__FILE__,__LINE__,(errp), \
       OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data)));
```

```
/* New order */

struct newinstruct {
   int w_id;
   int d_id;
   int c_id;
   int ol_i_id[15];
   int ol_supply_w_id[15];
   int ol_quantity[15];
};

struct newoutstruct {
   int terror;
   int o_id;
   int o_ol_cnt;
   char c_last[17];
   char c_credit[3];
   float c_discount;
   float w_tax;
   float d_tax;
   char o_entry_d[20];
   float total_amount;
   char i_name[15][25];
   int s_quantity[15];
   char brand_generic[15];
   float i_price[15];
   float ol_amount[15];
   char status[26];
   int retry;
};

struct newstruct {
   struct newinstruct newin;
   struct newoutstruct newout;
};


/* Payment */

struct payinstruct {
   int w_id;
   int d_id;
   int c_w_id;
   int c_d_id;
   int c_id;
   int bylastname;
   int h_amount;
   char c_last[17];
};

struct payoutstruct {
   int terror;
   char w_street_1[21];
   char w_street_2[21];
   char w_city[21];
   char w_state[3];
   char w_zip[10];
   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;
   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[11];
   char c_credit[3];
   double c_credit_lim;
```

```
   float c_discount;
   double c_balance;
   char c_data[201];
   char h_date[20];
   int retry;
};

struct paystruct {
   struct payinstruct payin;
   struct payoutstruct payout;
};


/* Order status */

struct ordinstruct {
   int w_id;
   int d_id;
   int c_id;
   int bylastname;
   char c_last[17];
};

struct ordoutstruct {
   int terror;
   int c_id;
   char c_last[17];
   char c_first[17];
   char c_middle[3];
   double c_balance;
   int o_id;
   char o_entry_d[20];
   int o_carrier_id;
   int o_ol_cnt;
   int ol_supply_w_id[15];
   int ol_i_id[15];
   int ol_quantity[15];
   float ol_amount[15];
   char ol_delivery_d[15][11];
   int retry;
};

struct ordstruct {
   struct ordinstruct ordin;
   struct ordoutstruct ordout;
};


/* Delivery */

struct delinstruct {
   int w_id;
   int o_carrier_id;
   double qtime;
   int in_timing_int;
};

struct deloutstruct {
   int terror;
   int retry;
};

struct delstruct {
   struct delinstruct delin;
   struct deloutstruct delout;
};


/* Stock level */

struct stoinstruct {
   int w_id;
   int d_id;
   int threshold;
};

struct stooutstruct {
   int terror;
   int low_stock;
   int retry;
};

struct stostruct {
   struct stoinstruct stoin;
   struct stooutstruct stoout;
};

#ifdef ORAKIT
#define message printf
#endif

#endif
```

lib/tpcc.h

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 14:01:49 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/
#ifndef TPCC_INCLUDED
#define TPCC_INCLUDED
#include <values.h>


/* The auditor can define these 20 char strings to be anything */
#define DRIVER_AUDIT_STRING  "driver audit string"
#define CLIENT_AUDIT_STRING  "client audit string"

#ifdef DEBUG
#define debug printf
#else
#define debug (void)
#endif

#include <stdio.h>

typedef int ID;            /* All id's */
typedef double MONEY;      /* Large integer number of cents */
typedef char TEXT;         /* Add an extra byte for null terminator */
typedef double TIME;       /* Elapsed seconds from start of run (float?) */
typedef int COUNT;         /* integer numbers of things */
typedef double REAL;       /* real numbers */
typedef int LOGICAL;       /* YES or NO */
typedef struct {           /* days and seconds since Jan 1, 1900 */
    int day;               /* NULL represented by negative day */
    int sec;
    } DATE;

/* Macro to convert time of day to TIME */
#include <time.h>
extern struct timeval start_time;
#define elapsed_time(t) ( ((t)->tv_sec - start_time.tv_sec) +  \
                          ((t)->tv_usec - start_time.tv_usec) / 1000000.0 )

typedef enum {Num,Money,Text,Time,Real,Date} FIELD_TYPE;   /* screen field types */


/* Various TPCC constants */
#define W_ID_LEN        4
#define D_ID_LEN        2
#define C_ID_LEN        4
#define I_ID_LEN        6
#define OL_QTY_LEN      2
#define PMT_LEN         7
#define C_ID_LEN        4
#define C_LAST_LEN      16
#define CARRIER_LEN     2
#define THRESHOLD_LEN   2
#define DIST_PER_WARE   10
#define CUST_PER_DIST   3000
#define ORD_PER_DIST    3000
#define MAXITEMS        100000
#define MAX_DIGITS      3       /* # of digits of the NURand number selected
                                   to generate the customer last name */
```

```
#define MAXWAREHOUSE   2000     /* maximum # of warehouses - scaling factor */
#define LOADSEED       42 /* # of digits of the NURand number selected

/*************************************************************/
/* database identifiers and populations                      */
/*************************************************************/

  int no_warehouse;                         /*  scaling factor    */
  int no_item;                              /*  100000            */
  int no_dist_pw;                           /*      10            */
  int no_cust_pd;                           /*    3000            */
  int no_ord_pd;                            /*    3000            */
  int no_new_pd;                            /*     900            */
  int tpcc_load_seed;                       /*     900            */

/* fields to add to each transaction for acid testing */
#define ACID_STUFF     \
    char acid_txn[2];  \
    int  acid_timing;  \
    int  acid_action;  \
    FILE *acid_res

typedef struct {
    ID OL_SUPPLY_W_ID;
    ID OL_I_ID;
    TEXT I_NAME[24+1];
    COUNT OL_QUANTITY;
    COUNT S_QUANTITY;
    MONEY I_PRICE;
    char brand_generic;
    } neworder_item;

typedef struct {
    int status;
    LOGICAL all_local;
    ID W_ID;
    ID D_ID;
    ID C_ID;
    TEXT C_LAST[C_LAST_LEN+1];
    TEXT C_CREDIT[2+1];
    REAL C_DISCOUNT;
    COUNT O_OL_CNT;
    ID O_ID;
    TEXT O_ENTRY_D[20];  /* dates as text fields */
    REAL W_TAX;
    REAL D_TAX;
    neworder_item item[15];
    ACID_STUFF;
    } neworder_trans;



typedef struct {
    int status;
    LOGICAL byname;
    ID W_ID;
    ID D_ID;
    ID C_ID;
    ID C_D_ID;
    ID C_W_ID;
    MONEY H_AMOUNT;
    TEXT H_DATE[20];  /* date as text field */
    TEXT W_STREET_1[20+1];
    TEXT W_STREET_2[20+1];
    TEXT W_CITY[20+1];
    TEXT W_STATE[2+1];
    TEXT W_ZIP[9+1];
    TEXT D_STREET_1[20+1];
    TEXT D_STREET_2[20+1];
    TEXT D_CITY[20+1];
    TEXT D_STATE[2+1];
    TEXT D_ZIP[9+1];
    TEXT C_FIRST[16+1];
    TEXT C_MIDDLE[2+1];
    TEXT C_LAST[16+1];
    TEXT C_STREET_1[20+1];
    TEXT C_STREET_2[20+1];
    TEXT C_CITY[20+1];
    TEXT C_STATE[2+1];
    TEXT C_ZIP[9+1];
    TEXT C_PHONE[16+1];
    TEXT C_SINCE[20]; /* date as text field */
    TEXT C_CREDIT[2+1];
    MONEY C_CREDIT_LIM;
    REAL C_DISCOUNT;
    REAL C_BALANCE;
    TEXT  C_DATA[200+1];
    ACID_STUFF;
    } payment_trans;


typedef struct {
    int status;
    LOGICAL byname;
    ID W_ID;
    ID D_ID;
    ID C_ID;
    TEXT C_FIRST[16+1];
    TEXT C_MIDDLE[2+1];
    TEXT C_LAST[16+1];
    MONEY C_BALANCE;
    ID O_ID;
    TEXT O_ENTRY_DATE[20];   /* date as text field */
    ID O_CARRIER_ID;
    COUNT ol_cnt;
    struct {
        ID OL_SUPPLY_W_ID;
        ID OL_I_ID;
        COUNT OL_QUANTITY;
        MONEY OL_AMOUNT;
        TEXT OL_DELIVERY_DATE[20]; /* date as text field */
        } item[15];
    ACID_STUFF;
    } ordstat_trans;


typedef struct {
    int status;
    ID W_ID;
    ID D_ID;
    COUNT threshold;
    COUNT low_stock;
    ACID_STUFF;
    } stocklev_trans;

typedef struct {
    int status;
    ID W_ID;
    ID O_CARRIER_ID;
    struct {
        ID O_ID;
          int status;
        } order[10];
    struct timeval enque[1];
    struct timeval deque[1];
    struct timeval complete[1];
    ACID_STUFF;
    } delivery_trans;


typedef union {
    neworder_trans neworder;
    payment_trans payment;
    ordstat_trans ordstat;
    delivery_trans delivery;
    stocklev_trans stocklev;
    int status;
    } generic_trans;


/*******************************************************
Record formats for results
*******************************************************/

#ifdef NOTYET
typedef struct
    {
    float t1, t2, t3, t4, t5;
```

```
    int status          :8;
    unsigned int type    :3;
    unsigned int ol_cnt  :4;
    unsigned int remote_ol_cnt :4;
    unsigned int byname :1;
    unsigned int remote  :1;
    unsigned int skipped  :4;
    } success_t;
#endif

typedef struct
    {
    TIME t1, t2, t3, t4, t5;
    int status;
    unsigned int type    :3;
    unsigned int ol_cnt  :4;
    unsigned int remote_ol_cnt :4;
    unsigned int byname :1;
    unsigned int remote  :1;
    unsigned int skipped  :4;
    } success_t;

typedef struct
    {
    struct timeval start_time;
    } success_header_t;


/*************************************************************************
Record formats for loading routines.  (DB's have own internal formats
*************************************************************************/
typedef struct
    {
    ID W_ID;
    TEXT W_NAME[10+1];
    TEXT W_STREET_1[20+1];
    TEXT W_STREET_2[20+1];
    TEXT W_CITY[20+1];
    TEXT W_STATE[2+1];
    TEXT W_ZIP[9+1];
    REAL W_TAX;
    MONEY W_YTD;
    } warehouse_row;

typedef struct
    {
    ID D_ID;
    ID D_W_ID;
    TEXT D_NAME[10+1];
    TEXT D_STREET_1[20+1];
    TEXT D_STREET_2[20+1];
    TEXT D_CITY[20+1];
    TEXT D_STATE[2+1];
    TEXT D_ZIP[9+1];
    REAL D_TAX;
    MONEY D_YTD;
    ID D_NEXT_O_ID;
    } district_row;


typedef struct
    {
    ID C_ID;
    ID C_D_ID;
    ID C_W_ID;
    TEXT C_FIRST[16+1];
    TEXT C_MIDDLE[2+1];
    TEXT C_LAST[16+1];
    TEXT C_STREET_1[20+1];
    TEXT C_STREET_2[20+1];
    TEXT C_CITY[20+1];
    TEXT C_STATE[2+1];
    TEXT C_ZIP[9+1];
    TEXT C_PHONE[16+1];
    DATE C_SINCE;
    TEXT C_CREDIT[2+1];
    MONEY C_CREDIT_LIM;
    REAL C_DISCOUNT;
    MONEY C_BALANCE;
    MONEY C_YTD_PAYMENT;
    COUNT C_PAYMENT_CNT;
    COUNT C_DELIVERY_CNT;
    TEXT  C_DATA[500+1];
    } customer_row;

typedef struct
    {
    ID H_C_ID;
    ID H_C_D_ID;
    ID H_C_W_ID;
    ID H_D_ID;
    ID H_W_ID;
    DATE H_DATE;
    MONEY H_AMOUNT;
    TEXT H_DATA[24+1];
    } history_row;

typedef struct
    {
    ID NO_O_ID;
    ID NO_D_ID;
    ID NO_W_ID;
    } neworder_row;

typedef struct
    {
    ID O_ID;
    ID O_D_ID;
    ID O_W_ID;
    ID O_C_ID;
    DATE O_ENTRY_D;
    ID O_CARRIER_ID;
    COUNT O_OL_CNT;
    LOGICAL O_ALL_LOCAL;
    } order_row;

typedef struct
    {
    ID OL_O_ID;
    ID OL_D_ID;
    ID OL_W_ID;
    ID OL_NUMBER;
    ID OL_I_ID;
    ID OL_SUPPLY_W_ID;
    DATE OL_DELIVERY_D;
    COUNT OL_QUANTITY;
    MONEY OL_AMOUNT;
    TEXT OL_DIST_INFO[24+1];
    } orderline_row;

typedef struct
    {
    ID I_ID;
    ID I_IM_ID;
    TEXT I_NAME[24+1];
    MONEY I_PRICE;
    TEXT I_DATA[50+1];
    } item_row;

typedef struct
    {
    ID S_I_ID;
    ID S_W_ID;
    COUNT S_QUANTITY;
    TEXT S_DIST_01[24+1];
    TEXT S_DIST_02[24+1];
    TEXT S_DIST_03[24+1];
    TEXT S_DIST_04[24+1];
    TEXT S_DIST_05[24+1];
    TEXT S_DIST_06[24+1];
    TEXT S_DIST_07[24+1];
    TEXT S_DIST_08[24+1];
    TEXT S_DIST_09[24+1];
    TEXT S_DIST_10[24+1];
    COUNT S_YTD;
    COUNT S_ORDER_CNT;
    COUNT S_REMOTE_CNT;
    TEXT S_DATA[50+1];
```

```
    } stock_row;


/* Empty field values */
#define EMPTY_NUM (MAXINT-1)
#define INVALID_NUM (MAXINT)
#define EMPTY_FLT   (MAXDOUBLE)
#define INVALID_FLT (MINDOUBLE)

/* Status conditions */
#define OK 0
#define E 1
#define E_INVALID_ITEM 2
#define E_NOT_ENOUGH_ORDERS 3
#define E_DB_ERROR 4
#define E_INVALID_INPUT 5

/* Error message strings */
static char *e_mesg[]={"Transaction complete.","Error","Invalid item number.",
                       "Not enough orders.", "Database ERROR !!!!"};



#define YES 1
#define NO 0



double cvt_flt();
double cvt_money();
TIME getclock();
TIME getlocalclock();

#define TPC_MSG_QUE  150


/*************************************
Transaction specific stuff
*************************************/

/* types of transactions */
#define NEWORDER 1
#define PAYMENT  2
#define ORDSTAT  3
#define DELIVERY 4
#define STOCKLEV 5
#define DEFERRED 6   /* deferred portion of delivery */

/* the name of each transaction */
static char *transaction_name[] =
    {"", "New_Order", "Payment", "Order-Status",
          "Delivery", "Stock-Level", "Deferred-Delivery"};

/* size of each transaction record */
static int transaction_size[] = {0,
     sizeof(neworder_trans),
     sizeof(payment_trans),
     sizeof(ordstat_trans),
     sizeof(delivery_trans),
     sizeof(stocklev_trans),
     sizeof(delivery_trans),
     0};

/* valid response time for each transaction */
static TIME valid_response[] = {0, 5, 5, 5, 5, 20};


#endif /* TPCC_INCLUDED */
```

lib/errlog.c

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 10:56:52 $
 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/
#include <stdio.h>
#include <varargs.h>
#include <unistd.h>
#include <errno.h>
#include <stdlib.h>
#include <fcntl.h>

int userid;

static msg_buf();

error(format, va_alist)
/*********************************************************************
error formats a message and outputs it to a standard location (stderr for now)
*********************************************************************/
    char *format;
          va_dcl
    {
    va_list argptr;

    msg_buf("error \n", strlen("error \n"));

    /* point to the list of arguments */
    va_start(argptr);

    /* format and print to stderr */
    vmessage(format, argptr);

    /* done */
    va_end(argptr);

    /* take an error exit */
    exit(1);
    }


syserror( format, va_alist )
/*********************************************************************
syserror logs a message with the system error code
*********************************************************************/
    char *format;
          va_dcl
    {
    va_list argptr;
    int save_errno = errno;

    msg_buf("syserror \n", strlen("syserror \n"));
    /* point to the list of arguments */
    va_start(argptr);

    /* format and print to stderr */
    vmessage(format, argptr);

    /* done */
    va_end(argptr);

    /* display the system error message */
    message("    System error message: %d %s\n", save_errno, strerror(save_errno));

    /* take an error exit */
    exit(1);
    }



message(format, va_alist)
/*********************************************************************
message formats a message and outputs it to a standard location (stderr for now)
*********************************************************************/
    char *format;
          va_dcl
{
    va_list argptr;

    msg_buf("message \n", strlen("message \n"));
    /* point to the list of arguments */
```

```
    va_start(argptr);

    /* format and print to stderr */
    vmessage(format, argptr);

    /* done */
    va_end(argptr);
    }



vmessage(format, argptr)
/************************************************

************************************************/
    char *format;
    va_list argptr;
{
  char buf[3*1024];

  /* format a message id */
  sprintf(buf, "Host %-8s User %-6d Pid %-6d ", getenv("HOST_NAME"), userid,
getpid());

  /* format the string and print it */
  vsprintf(buf+strlen(buf), format, argptr);
  if (getenv("NO_ERROR_LOG") == NULL)
      msg_buf(buf, strlen(buf));
  if (getenv("NO_STDERR") == NULL)
      write(2, buf, strlen(buf));
}



static msg_buf(buf, size)
  char *buf;
  int size;
  {
  int     fd;
  char  *fname;
  time_t tepoch = time(NULL);
  char timestamp[16];
  int ltimestamp;

  ltimestamp = strftime(timestamp, sizeof(timestamp), "%m/%d %T ",
localtime(&tepoch));

  /* get the file name to use */
  fname = getenv("ERROR_LOG");
  if (fname == NULL)
    fname = "/tmp/ERROR_LOG";

  /* get exclusive access to the error log file */
  fd= open(fname, O_WRONLY | O_CREAT, 0666);
  if (fd < 0)
    console_error("Can't open tpc error log file 'ERROR_LOG'\n");
  lockf(fd, F_LOCK, 0);

  /* write the new text at the end of the file */
  lseek(fd, 0, SEEK_END);
  write(fd, timestamp, ltimestamp);
  write(fd, buf, size);

  /* release the file */
  /* fsync(fd); */
  lockf(fd, F_ULOCK, 0);
  close(fd);
  }



console_error(str)
   char *str;
   {
   int fd = open("/dev/tty", O_WRONLY);
   write(fd, str, strlen(str));
   close(fd);
   exit(1);
   }
```

lib/fmt.c

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 10:56:52 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/
#include "tpcc.h"
#include "iobuf.h"
#include <math.h>    /* needed for ceil (VM) */
#include <strings.h>

/* formatting routines.  */

/* Note: Currently use integer routines to format and convert.  Need to
   modify the code for cases when integers don't work.  */

fmt_money(str, m, width)
    char *str;
    MONEY m;
    int width;
    {

    if (m == EMPTY_FLT)
        {
        memset(str, '_', width);
        str[width] = '\0';
        return;
        }

    /* format it as a number with a leading blank */
    *str = ' ';
    fmt_flt(str+1, m/100, width-1, 2);

    /* fill in a leading dollar */
    while (*(str+1) == ' ')
        str++;
    *str = '$';
    }


double cvt_money(str)
    char *str;
    {
    char temp[81], *t, *s;
    double cvt_flt(), f;

    /* skip leading and trailing blanks */
    cvt_text(str, temp);

    /* remove leading $ */
    if (*temp == '$')   t = temp + 1;
    else                t = temp;

    /* start scan at current character */
    s = t;

    /* allow leading minus sign */
    if (*s == '-')
        s++;

    /* allow leading digits */
    while (isdigit(*s))
        s++;

    /* allow decimal pt and two decimal digits */
    if (*s == '.')    s++;
    if (isdigit(*s))  s++;
    if (isdigit(*s))  s++;

    /* There should be no more characters */
    if (*s != '\0') return INVALID_FLT;
```

```
    /* convert the floating pt number */
    f = cvt_flt(t);
    if (f == EMPTY_FLT)         return EMPTY_FLT;
    else if (f == INVALID_FLT) return INVALID_FLT;
    else                        return rint(f*100);
    }


fmt_num(str, n, width)
    char str[];
    int n;
    int width;
    {
    /* mark the end of the string */
    str[width] = '\0';

    /* if empty number, return the empty field */
    if (n == EMPTY_NUM)
        memset(str, '_', width);

    /* otherwise, convert the integer */
    else
        fmtint(str, n, width, ' ');

    debug("fmt_num:  n=%d  str=%s\n", n, str);
    }



cvt_num(str)
    char str[];
    {
    char text[81];
    cvt_text(str, text);
    if (*text == '\0')
        return EMPTY_NUM;
    else
        return cvtint(text);
    }



fmt_flt(str, x, width, dec)
/*************************************************************************
fmt_flt converts a floating pt number to a string   "999999.9999"
*************************************************************************/
    char *str;
    double x;
    int width;
    int dec;
    {
    int negative;
    int integer, fract;
    double absolute;

    static double pow10[] =
    {1., 10., 100., 1000., 10000., 100000., 1000000., 10000000., 100000000.};

    /* mark the end of string */
    str[width] = '\0';

    /* if empty value, make it be an empty field */
    if (x == EMPTY_FLT)
        {
        memset(str, '_', width);
        return;
        }

    absolute = (x < 0)? -x: x;

    /* separate into integer and fractional parts */
    integer = (int) absolute;
    fract   = (absolute - integer) * pow10[dec] + .5;

    /* let the integer portion contain the sign */
    if (x < 0) integer = -integer;

    /* Format integer and fraction separately */
    fmtint(str, integer, width-dec-1, ' ');
    str[width-dec-1] = '.';
    fmtint(str+width-dec, fract, dec, '0');
    }



double cvt_flt(str)
    char str[];
    {
    char text[81];
    char *t;
    double value;
    int div;
    int fract;
    int negative;
    int i;

    /* normalize the text */
    cvt_text(str, text);
    if (*text == '\0')
        return EMPTY_FLT;

    negative = NO;
    fract = NO;
    value = 0;
    div = 1.0;

    negative = (text[0] == '-');
    if (negative)    t = text+1;
    else             t = text;

    for (;  *t !=  '\0' ; t++)
        {

        if (*t == '.')
              if (fract)  return INVALID_FLT;
            else        fract = YES;

        else if (isdigit(*t))
            {
            value = value*10 + (int)*t - (int)'0';
            if (fract) div *= 10;
            }

        else
            return INVALID_FLT;
        }

    if (fract)
        value /= div;

    if (negative)
        value = -value;

    return value;
    }



fmt_text(s, text, width)
    char *s, *text;
    int width;
    {

    /* if an empty string, then all underscores */
    if (*text == '\0')
        for (; width > 0; width--)
            *s++  = '_';

    /* otherwise, blank fill it */
    else
        {

        /* copy the text into the new buffer */
        for ( ; *text != '\0';  width--)
            *s++ = *text++;
```

```
        /* fill in the rest with blanks */
        for (;  width > 0;  width--)
            *s++ = ' ';
        }

    /* and finally, terminate the string */
    *s = '\0';
    }


cvt_text(s, text)
    char *s;
    char *text;
    {
    char *lastnb;

    /* skip leading blanks and underscores */
    for (; *s == ' ' || *s == '_'; s++)
        ;

    /* copy the characters, keeping track of last blank or underscore */
    lastnb = text-1;
    for (; *s != '\0'; *text++ = *s++)
        if (*s != ' ' && *s != '_')
            lastnb = text;

    /* truncate the text string to last nonblank character */
    *(lastnb+1) = '\0';
    }


fmtint(field, value, size, fill)
/*****************************************************************
fmtint formats an integer value into a character field to make the integer
right-justified within the character field, padded with leading fill
characters (e.g. leading blanks if a blank is passed in for the fill argument
*****************************************************************************/
    int value;
    char *field;
    int size;
    char fill;
    {
    int negative;
    int dividend;
    int remainder;
    char *p;

    /* create characters from right to left */
    p = field + size - 1;

    /* make note if this is a negative number */
    negative = value < 0;
    if (negative)
        value = -value;

    /* Case: Null field.  Can't do anything */
    if (p < field)
        ;

    /* Case: value is zero.  Print a leading '0' */
    else if (value == 0)
        *p-- = '0';

    /* Otherwise, convert each digit in turn */
    else do
        {

        dividend = value / 10;
        remainder = value - dividend * 10;
        value = dividend;

        *p-- = (char) (  (int)'0'  +  remainder  );

        } while (p >= field && value > 0);

    /* insert a minus sign if appropriate */
    if (negative && p >= field)
        *p-- = '-';

    /* fill in leading characters */
    while (p >= field)
        *p-- = fill;
    }


int cvtint(str)
/*************************************************************
getint extracts an integer value from the given character field
(ex: turns the string "123" into the integer 123)
*************************************************************/
    char *str;
    {
    int value;
    char c;
    int negative;
    debug("cvtint: str=%s\n", str);

    negative = (*str == '-');
    if (negative)  str++;

    /* convert the integer */
    for (value = 0; isdigit(*str); str++)
        value = value*10 + (int)(*str) - (int)'0';

    /* if any non-digit characters, error */
    if (*str != '\0')
       return INVALID_NUM;

    /* make negative if there was a minus sign */
    if (negative)
        value = -value;

    debug("cvtint: value=%d\n", value);
    return value;
    }


fmt_phone(str, phone)
    char str[20];
    char *phone;

    {
    /* copy phone number and insert dashes  999999-999-999-9999 */
    str[0] = phone[0];  str[1] = phone[1]; str[2] = phone[2];
    str[3] = phone[3];  str[4] = phone[4]; str[5] = phone[5];
    str[6] = '-';
    str[7] = phone[6];  str[8] = phone[7]; str[9] = phone[8];
    str[10] = '-';
    str[11] = phone[9]; str[12] = phone[10]; str[13] = phone[11];
    str[14] = '-';
    str[15] = phone[12]; str[16] = phone[13]; str[17] = phone[14];
    str[18] = phone[15];
    str[19] = '\0';
    }


fmt_zip(str,zip)
    char str[20];
    char *zip;
    {
    /* copy zip code and insert dashes 99999-9999 */
    str[0] = zip[0];  str[1] = zip[1];  str[2] = zip[2];
    str[3] = zip[3];  str[4] = zip[4];
    str[5] = '-';
    str[6] = zip[5]; str[7] = zip[6]; str[8] = zip[7];  str[9] = zip[8];
    str[10] = '\0';
    }
```

lib/iobuf.h

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

```
 @(#) Version: A.10.10 $Date: 97/12/15 10:56:52 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*************************************************************************/

/*********************************************************************
History
941220 LAN Added definition and initialization of the line_col[] array.
       This was needed for modifications made of client program to do
       block I/O using a WYSE terminal.
*********************************************************************/

/* structure for screen emulation */
typedef struct
    {
    int row;
    int col;
    char buf[25][81];
    } screen_t;


typedef struct {
    char *beg;
    char *end;   /* for output buffers */
    char *max;
    char *cur;   /* for input buffers */
    } iobuf;

/* Macro do define an I/O buffer of x characters, initialized to empty */
#define define_iobuf(name, size)                          \
char name##_data[size];                                  \
iobuf name[1] = {{name##_data, name##_data,       \
                 name##_data+size, name##_data}}

#define reset(buf)  if (1) { \
    (buf)->cur = (buf)->end = (buf)->beg;                \
    *(buf)->beg = '\0';                                   \
    } else (void)0

#define flush()  if(1) { \
        display(out_buf);    \
        reset(out_buf);     \
     } else (void)0


/* Standard I/O to and from in_buf and out_buf */
#ifdef DECLARE_IO_BUFFERS
define_iobuf(output_stuff, 4*1024);
define_iobuf(input_stuff, 1024);
iobuf *in_buf = input_stuff;
iobuf *out_buf = output_stuff;
#else
iobuf *in_buf;
iobuf *out_buf;
#endif

#define pushc(c) if (1) { \
    if (out_buf->end >= out_buf->max)   \
         error("out_buf overflow:  beg=0x%x end=%d  max=%d\n",  \
           out_buf->beg, out_buf->end-out_buf->beg,out_buf->max-out_buf->beg); \
    *(out_buf->end++) = (c);  \
    *(out_buf->end) = '\0';  /* debug */ \
    } else (void)0

#define popc()  \
    (*in_buf->cur++)


/* Standard characters used for screen control */
#define ENTER '\015'
#define TAB   '\t'
#define BACKTAB '\02'  /*  ^B  */
#define CNTRLC '\03'
#define BACKSPACE '\010'
#define BELL '\07'
#define BLANK ' '
#define UNDERLINE '_'
#define ESCAPE '\033'
/*#define EOF    ((char)-1) */
#define TRIGGER '\021'  /* dc1 */
```

lib/iobuf.c

```
/*************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 10:56:52 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*************************************************************************/
#define DECLARE_IO_BUFFERS
#include "iobuf.h"
#undef DECLARE_IO_BUFFERS
#include "tpcc.h"
#include <errno.h>


string(str)
    char str[];
    {
    for (; *str != '\0'; str++)
        pushc(*str);
    }


push(str, len)
    char *str;
    int len;
    {
    for (; len > 0; len --)
        pushc(*str++);
    }


display(scr)
    iobuf *scr;
    {
/* Note: if problems doing output, let the input routine detect it */
    char *p;
    int len;
    for (p = scr->beg; p < scr->end; p+=len)
        {
        len = write(1, p, scr->end - p);
        if (len <= 0) break;
        }
    }


input(scr)
    iobuf *scr;
    {
    int len;

    /* read in as many characters as are available */
    len = read(0, scr->end, scr->max - scr->end);

    /* if end of input, then pretend we read an END_character */
    if (len == 0 || (len == -1 && errno == ECONNRESET))
        {
        *scr->end = EOF;
        len = 1;
        }

    /* Check for errors */
    else if (len == -1)
        syserror("input(scr): unable to read stdin\n");

    /* update the pointers to reflect the new data */
    scr->end += len;
    *scr->end='\0';  /* for debugging */
    }
```

```
getkey()
    {
    if (in_buf->cur == in_buf->end)
        {
        flush();
        reset(in_buf);
        input(in_buf);
        }

    return popc();
    }
```

lib/random.c

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 14:01:59 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/
#include "tpcc.h"
#include "string.h"
#include "random.h"

double drand48();

char lastNames[1000][16];
char customerData1[10][301];
char customerData2[10][201];
char stockData1[10][27];
char stockData2[10][25];
char historyData1[10][13];
char historyData2[10][13];
char citystreetData1[10][11];
char citystreetData2[10][11];
char firstNameData1[10][9];
char firstNameData2[10][9];
char StockDistrict[10][25];
char phoneData[10][17];

static long RandySeedIter = 7;

void GenerateLastNames()
{
    int      i;
    char *name;
    static char *n[] = {"BAR", "OUGHT", "ABLE", "PRI", "PRES",
                        "ESE", "ANTI", "CALLY", "ATION", "EING"};

    for(i = 0; i < 1000; i++) {
        name = lastNames[i];
        strcpy(name, n[(i/100)%10]);
        strcat(name, n[(i/10) %10]);
        strcat(name, n[(i/1)  %10]);
    }
}

int MakeNumberString(min, max, num)
    int min;
    int max;
    TEXT num[];
    {
    static char digit[]="0123456789";
    int length;
    int i;

    length = RandomNumber(min, max);

    for (i=0;  i<length;  i++)
        num[i] = digit[RandomNumber(0,9)];
    num[length] = '\0';

    return length;
    }
```

```
ID RandomWarehouse(local, scale, percent)
    ID local;
    ID scale;
    int percent;   /* percent of remote transactions */
    {
    ID w_id;

    /* For the given percent of the time, pick the local warehouse */
    if (RandomNumber(1, 100) > percent || scale == 1)
        w_id = local;

    /* Otherwise, pick a non-local warehouse */
    else
        {
        w_id = RandomNumber(2, scale);
        if (w_id == local)
            w_id = 1;
        }
    return w_id;
    }


/* Initialize a table of Random strings for the stock-district
   field in the stock table.  We can use a table of 10 elements
   and select randomly from this table via rule 4.3.2.2 in
   the TPC-C spec */
void InitRandomStrings()
{
    int i;

    for (i=0; i < 10; i++) {
              MakeAlphaString(24,24,&StockDistrict[i]);

              MakeAlphaString(300,300,&customerData1[i]);
              MakeAlphaString(0,200,&customerData2[i]);

              MakeAlphaString(26,26,&stockData1[i]);
              MakeAlphaString(0,24,&stockData2[i]);

              MakeAlphaString(12,12,&historyData1[i]);
              MakeAlphaString(0,12, &historyData2[i]);

              MakeAlphaString(10,10,&citystreetData1[i]);
              MakeAlphaString(0 ,10,&citystreetData2[i]);

              MakeAlphaString(8,8,&firstNameData1[i]);
              MakeAlphaString(0,8,&firstNameData2[i]);

              MakeNumberString(16,16,&phoneData[i]);
         }
         GenerateLastNames();
}

int MakeAlphaString(min, max, str)
    int min;
    int max;
    TEXT str[];
    {
    static char character[] = "abcdefghijklmnopqrstuvwxyz";
    int length;
    int i;

    length = RandomNumber(min, max);

    for (i=0;  i<length;  i++) {
        /* NOTE:  we use sizeof(character)-2 because of the following:
           subtract 1 because we are numbering from 0 instead of 1 and
           subtract 1 because the sizeof(character) is 1 greater than
           the data in character because of the inivisible C string
           terminator at the end. */
        str[i] = character[RandomNumber(0, sizeof(character)-2)];
    }
    str[length] = '\0';

    return length;
    }

void RandomPermutation(perm, n)
    int perm[];
```

```
    int n;
    {
    int i, r, t;

    /* generate the identity permutation to start with */
    for (i=1; i<=n; i++)
        perm[i] = i;

    /* randomly shuffle the permutation */
    for (i=1; i<=n; i++)
        {
        r = RandomNumber(i, n);
        t = perm[i]; perm[i] = perm[r]; perm[r] = t;
        }
    }


void RandomDelay(mean, adjust)
/*****************************************************************
random_sleep sleeps according to the TPC specification
*****************************************************************/
    double mean;
    double adjust;
    {
    double secs;
    double exponential();

    secs = exponential(mean);

    delay(secs+adjust);
    }

double exponential(mean)
/*****************************************************************
exponential generates a reverse exponential distribution
*****************************************************************/
    double mean;
    {
    double x;
    double log();

#ifdef USE_DRAND48
    x = -log(1.0-drand48()) * mean;
#else
    x = -log(1.0-randy()) * mean;
#endif

    return x;
    }

void SetRandomSeed(val)
long val;
    {
#ifdef USE_DRAND48
    srand48(val);
#else
    RandySeedIter = val;
#endif
    }

void Randomize()
    {
    SetRandomSeed(time(0)+getpid());
    }

/* Random number generator from Proceeding of the ACM */
#define RANDY_A_VAL 16807
/* 2^31 - 1 */
#define RANDY_M_VAL 2147483647
/* m / a */
#define RANDY_Q_VAL 127773
/* m % a */
#define RANDY_R_VAL 2836

double randy()
{
  long    hi, lo, test;

  hi = RandySeedIter / RANDY_Q_VAL;

  lo = RandySeedIter % RANDY_Q_VAL;

  test = (RANDY_A_VAL * lo) - (RANDY_R_VAL * hi);
  RandySeedIter = (test > 0) ? test : test + RANDY_M_VAL;

  return( (double)RandySeedIter / (double)RANDY_M_VAL );
}  /* end of fn randy  */
```

lib/random.h

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 14:02:00 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/
#ifndef TPCC_RANDOM
#define TPCC_RANDOM

#ifdef USE_DRAND48
double drand48();
#else
double randy();
#endif

extern int    MakeNumberString();
extern ID     RandomWarehouse();
extern int    MakeAlphaString();
extern void   RandomPermutation();
extern void   RandomDelay();
extern double exponential();
extern void   Randomize();
extern void   SetRandomSeed();

extern char lastNames[1000][16];
extern char customerData1[10][301];
extern char customerData2[10][201];
extern char stockData1[10][27];
extern char stockData2[10][25];
extern char historyData1[10][13];
extern char historyData2[10][13];
extern char citystreetData1[10][11];
extern char citystreetData2[10][11];
extern char firstNameData1[10][9];
extern char firstNameData2[10][9];
extern char StockDistrict[10][25];
extern char phoneData[10][17];

/*****************************************************************************
RandomNumber selects a uniform random number from min to max inclusive
*****************************************************************************/
#ifdef USE_DRAND48
#define RandomNumber(min,max)  \
          ((int)(drand48() * ((int)(max) - (int)(min) + 1)) + (int)(min))
#else
#define RandomNumber(min,max)  \
          ((int)(randy() * ((int)(max) - (int)(min) + 1)) + (int)(min))
#endif

/*****************************************************************************
NURandomNumber selects a non-uniform random number
*****************************************************************************/
#define NURandomNumber(a, min, max, c) \
    ((RandomNumber(0, a) | RandomNumber(min, max)) + (c)) % \
                    ((max) - (min) + 1) + (min)


#define SelectCityStreetData(data) \
{ \
    strcpy(data,citystreetData1[RandomNumber(0,9)]); \
    strcat(data,citystreetData2[RandomNumber(0,9)]); \
}

#define SelectFirstName(data) \
{ \
    strcpy(data,firstNameData1[RandomNumber(0,9)]); \
    strcat(data,firstNameData2[RandomNumber(0,9)]); \
```

```
}

#define SelectHistoryData(data) \
{ \
    strcpy(data,historyData1[RandomNumber(0,9)]); \
    strcat(data,historyData2[RandomNumber(0,9)]); \
}

#define SelectStockData(data) \
{ \
    strcpy(data,stockData1[RandomNumber(0,9)]); \
    strcat(data,stockData2[RandomNumber(0,9)]); \
}

#define SelectClientData(data) \
{ \
    strcpy(data,customerData1[RandomNumber(0,9)]); \
    strcat(data,customerData2[RandomNumber(0,9)]); \
}

#define SelectPhoneData(data) strcpy(data,phoneData[RandomNumber(0,9)])
#define SelectStockDistrict(data) strcpy(data,StockDistrict[RandomNumber(0,9)])

#define MakeZip(zip) \
{ \
   MakeNumberString(4, 4, zip); \
   zip[4] = '1'; \
   zip[5] = '1'; \
   zip[6] = '1'; \
   zip[7] = '1'; \
   zip[8] = '1'; \
   zip[9] = '\0'; \
}

#define MakeAddress(str1, str2, city, state, zip) \
{ \
    SelectCityStreetData(str1); \
    SelectCityStreetData(str2); \
    SelectCityStreetData(city); \
    MakeAlphaString(2,2,state); \
    MakeZip(zip); \
}

#define LastName(num, name) strcpy(name, lastNames[(num)])

#define Original(str) \
{ \
    int len = strlen(str); \
    if (len >= 8) { \
        int pos = RandomNumber(0,(len-8)); \
        str[pos+0] = 'O'; \
        str[pos+1] = 'R'; \
        str[pos+2] = 'I'; \
        str[pos+3] = 'G'; \
        str[pos+4] = 'I'; \
        str[pos+5] = 'N'; \
        str[pos+6] = 'A'; \
        str[pos+7] = 'L'; \
    } \
}


#endif
```

lib/Makefile

```
#*******************************************************************************
#@(#) Version: A.10.10 $Date: 98/02/03 08:13:40 $
#
#(c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
#*******************************************************************************

include ../buildenv.mk

#
# Makefile for compiling the client, batch-tpcc, and service code
#

OH      = ${ORACLE_HOME}
OL      = ${ORACLE_HOME}/lib
ORL     = ${ORACLE_HOME}/rdbms/lib
P       = ${WORK_DIR}/src
I       = $(P)/lib
L       = $(P)/lib
D       = $(P)/driver
Q       = $(P)/que
S       = $(P)/client

##include $(ORACLE_HOME)/bench/buildtools/prefix.mk

SH_OPT     = -Wl,-a,shared
OPT        =  -Wl,-a,archive_shared
LDOPTS     = -ldld -a archive_shared

TUXEDO     = -D_HPUX_SOURCE ${ROOTDIR}/include ${OPT}

ORA_CFLAGS = -DORA_BLOCK_PATH=\"/project/tpcc/blocks/\" \
             -DORA_NULL_DATE=\"01-01-1811\" \
             -D_HPUX_SOURCE -DSS_64BIT_SERVER -DHPPA64 -DSLS8NATIVE -DSLU8NATIVE \
             -DSLTS_ENABLE -DHPUX_KTHREAD -DSLXMX_ENABLE -DSLMXMX_ENABLE

ORA_INCLUDE=-I. -I${S}/oracle \
    -I$(ORACLE_HOME)/rdbms/demo \
    -I$(ORACLE_HOME)/rdbms/public -I$(ORACLE_HOME)/rdbms/include \
    -I$(ORACLE_HOME)/plsql/public \
    -I$(ORACLE_HOME)/network/public
SYB_INCLUDE= -I${SYBASE}/include
VIS_INCLUDE= -I ${VISIGENIC}/include
TUX_INCLUDE= -I${ROOTDIR}/include
INCLUDE    = -I. -I$L

SH_CFLAGS  = $(BUILDFLAGS) ${SH_OPT} ${INCLUDE} ${TUX_INCLUDE}
CFLAGS       = $(BUILDFLAGS) ${OPT} ${INCLUDE} ${TUX_INCLUDE}
CFLAGS_SYB = $(BUILDFLAGS) ${OPT} ${INCLUDE} ${TUX_INCLUDE} ${SYB_INCLUDE}
CFLAGS_ORA = $(BUILDFLAGS) ${ORA_CFLAGS} ${OPT} ${INCLUDE} ${ORA_INCLUDE}
${TUX_INCLUDE}
CFLAGS_SQL = -Dunix -D_HPUX_SOURCE -DVG_UNIX ${OPT} ${INCLUDE} ${TUX_INCLUDE}
${SQL_INCLUDE} ${VIS_INCLUDE}

ORA_LOAD =  -L${OL} -L${ORL} \
${ORL}/ssdbaed.o ${ORL}/kpudfo.o \
${OL}/nautab.o ${OL}/naeet.o ${OL}/naect.o ${OL}/naedhs.o \
-lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lnro8 -lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lclient8 -lvsn8 -lcommon8 -lskgxp8 -lgeneric8 \
-lmm \
-lnls8 -lcore8 -lnls8 -lcore8 -lnls8 \
-lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lnro8 -lnoss8 \
-lncrypt8 -lntcp8 -lntcps8 -lnsslb8 -lnus8 -ln8 -lntns8 -lnoname8 -lnhost8 -lnbeq8 -
lnldap8 -lnl8 \
-lclient8 -lvsn8 -lcommon8 -lskgxp8 -lgeneric8 \
-ltrace8 \
-lnls8 -lcore8 -lnls8 -lcore8 -lnls8 \
-lclient8 -lvsn8 -lcommon8 -lskgxp8 -lgeneric8 \
-lnls8 -lcore8 -lnls8 -lcore8 -lnls8 \
-lldapclnt8 \
-lcl -lm -lpthread -l:libc.sl -ldld

#`cat ${OL}/sysliblist`\
#

LDFLAGS_ORA= ${OPT} $(CFLAGS_ORA) ${L}/tpc_lib.a ${ORA_LOAD}
LDFLAGS_SYB= ${OPT} ${L}/tpc_lib.a -L${SYBASE}/lib -lsybdb -lm
LDFLAGS_SQL= ${OPT} ${L}/tpc_lib.a -L/opt/odbc/lib -lodbc -lm

PROGRAMS     = client service startup client_batch msg_server raw
```

```
install: others_oracle

all_ora: others_oracle service_oracle tpcc_client

tpcc_client: client
        $(MV) client ${WORK_DIR}/bin/
others_sybase: raw startup client_batch_syb msg_server_syb
        $(MV) raw startup client_batch msg_server ${WORK_DIR}/bin
others_oracle: raw startup client_batch_ora
        $(MV) raw startup client_batch ${WORK_DIR}/bin
others_sqlserver: raw startup client_batch_sql msg_server_sql
        $(MV) raw startup client_batch msg_server ${WORK_DIR}/bin
service_oracle: service_ora
        $(MV) service ${WORK_DIR}/bin/
service_sybase: service_syb
        $(MV) service ${WORK_DIR}/bin/
service_sqlserver: service_sql
        $(MV) service ${WORK_DIR}/bin/

${S}/sybase/transaction.o: ${S}/sybase/transaction.c
        $(CC) ${CFLAGS_SYB} $(L)/tpc_lib.a -c ${S}/sybase/transaction.c;
${S}/sqlserver/transactionb.o: ${S}/sqlserver/transactionb.c
        $(CC) ${CFLAGS_SQL} $(L)/tpc_lib.a -c ${S}/sqlserver/transactionb.c;

ORA_OBJS=plnew.o plord.o plpay.o pldel.o plsto.o tpccpl.o

transaction.o: ${S}/oracle/transaction.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/transaction.c;
plnew.o: ${S}/oracle/plnew.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plnew.c;
plord.o: ${S}/oracle/plord.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plord.c;
plpay.o: ${S}/oracle/plpay.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plpay.c;
pldel.o: ${S}/oracle/pldel.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/pldel.c;
plsto.o: ${S}/oracle/plsto.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/plsto.c;
tpccpl.o: ${S}/oracle/tpccpl.c
        $(CC) ${CFLAGS_ORA} $(L)/tpc_lib.a -c ${S}/oracle/tpccpl.c;

raw: raw.o
        cc ${CFLAGS} raw.o $(L)/tpc_lib.a -o raw

startup: startup.o $(L)/tpc_lib.a
        cc ${SH_CFLAGS} startup.o $(L)/tpc_lib.a -o startup
        chmod a+rw startup

# Warning: can't use +pd because kernel will use this size to extend
# DATA pregion when break/sbreak is called.
# Force shared libc to avoid to both libc data from the archived and shared version.
client: client.o  tux_transaction.o $(L)/tpc_lib.a
        cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
        ${ROOTDIR}/bin/buildclient -v -o client \
       -f "$(BUILDFLAGS) $(OPT) -Wl,+pi 256K -Wl,-R 0x100000 \
           client.o tux_transaction.o $(L)/tpc_lib.a" \
        -l "-lnsl  -lm -Wl,-ashared -lc"
        cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

service_ora: service.o transaction.o $(ORA_OBJS) $(L)/tpc_lib.a
        cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
        mv /project/iti/lib/libgp.a /project/iti/lib/libgp.a.cc_service
        ${ROOTDIR}/bin/buildserver -v -b shm \
                -s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC \
                -o service \
                -f "$(BUILDFLAGS) $(OPT) \
                    -Wl,+pi 4M -Wl,+pd 256K \
                    -Wl,-R 0x400000 -Wl,-D 0x40100000 \
                       service.o transaction.o $(ORA_OBJS) $L/tpc_lib.a \
                    ${LDFLAGS_ORA}" \
                -l "-lnsl "
        mv /project/iti/lib/libgp.a.cc_service /project/iti/lib/libgp.a
        cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

service_syb: service.o ${S}/sybase/transaction.o $(L)/tpc_lib.a
        cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
        ${ROOTDIR}/bin/buildserver -v -b shm \
                -s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC \
                -o service \
                -f "service.o transaction.o $L/tpc_lib.a \
                    ${SYBASE}/lib/libsybdb.a -lm";
        cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

service_sql: service.o ${S}/sqlserver/transactionb.o $(L)/tpc_lib.a
        cp /project/iti/udataobj/lic.txt.sd /project/iti/udataobj/lic.txt
        ${ROOTDIR}/bin/buildserver -v -b shm \
                -s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC \
                -o service \
                -f "service.o transactionb.o $L/tpc_lib.a \
                   /vsbuild/v1.10/build/com/obj/inst/libodbc.sl"
        cp /project/iti/udataobj/lic.txt.rt /project/iti/udataobj/lic.txt

client_batch_ora: $(D)/driver.o $(D)/generate.o transaction.o $(ORA_OBJS)
$(Q)/dummy_que.o $(L)/tpc_lib.a  \
                   $(L)/server_default.o
        $(CC) $(BUILDFLAGS) $(OPT) \
        -Wl,+pi 4M -Wl,+pd 256K \
        -Wl,-R 0x400000 -Wl,-D 0x40100000 \
        $(D)/driver.o $(D)/generate.o transaction.o \
        $(ORA_OBJS) $(Q)/dummy_que.o $(L)/server_default.o $(L)/tpc_lib.a \
        ${LDFLAGS_ORA} -o client_batch;

client_batch_syb: $(D)/driver.o $(D)/generate.o transaction.o $(Q)/dummy_que.o
$(L)/tpc_lib.a  \
                   $(L)/server_default.o
        $(CC) $(D)/driver.o $(D)/generate.o transaction.o $(Q)/dummy_que.o
$(L)/server_default.o $(L)/tpc_lib.a ${LDFLAGS_SYB} -o client_batch;

client_batch_sql: $(D)/driver.o $(D)/generate.o transactionb.o $(Q)/dummy_que.o
$(L)/tpc_lib.a \
                                        $(L)/server_default.o
                  $(CC) $(D)/driver.o $(D)/generate.o transactionb.o
$(Q)/dummy_que.o $(L)/server_default.o $(L)/tpc_lib.a ${LDFLAGS_SQL} -o
client_batch;

msg_server_ora: $(Q)/msg_server.o  transaction.o $(ORA_OBJS) $(L)/tpc_lib.a
        $(CC) $(Q)/msg_server.o transaction.o $(ORA_OBJS) ${LDFLAGS_ORA} -o
msg_server;
msg_server_syb: $(Q)/msg_server.o  transaction.o $(L)/tpc_lib.a
        $(CC) $(Q)/msg_server.o transaction.o ${LDFLAGS_SYB} -o msg_server;

msg_server_sql: $(Q)/msg_server.o  transactionb.o $(L)/tpc_lib.a
        $(CC) $(Q)/msg_server.o transactionb.o ${LDFLAGS_SQL} -o msg_server;

clean:
        rm -f *.o

clobber: clean
        rm -f ${PROGRAMS}
```

lib/tas.s

```
;*****************************************************************************
; @(#) Version: A.10.10 $Date: 97/12/15 10:56:55 $
;
; (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
;*****************************************************************************
    .code
;*****************************************************************
; Test and set routines implemented with the LDCWS instruction
;
; The LDCWS instruction is very similar to a traditional test+set instruction
; with the following exceptions:
;   o Zero means set and One means clear.
;   o Word must be 16 byte aligned
;
; Acheiving 16 byte alignment is awkward in C, so these routines have
; been designed to take a larger unaligned structure and round up to the
; first 16 byte aligned word of the structure.
;
; A reasonable C declaration for this structure could be:
;     typedef  struct { int words[4]; }  latch_t.
```

```
;
; On future multiprocessor  machines,  normal load and store instructions could
; be reordered arbitrarily by the hardware.  The stws,ma 0() and ldws,ma 0()
; instructions will force a synchronization of reordered loads and stores.
;*****************************************************************************



; tas(latch)
;***************************************************
; tas is true if we succeeded in acquiring the latch
;***************************************************
    .proc
    .callinfo
    .export tas
tas

; test+set, and return
    bv     (rp)
    ldcws  (arg0), ret0
    .procend
```

lib/results_file.c

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 14:02:01 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include "tpcc.h"



static FILE *rfile;

results_open(id)
    int id;
    {
    char fullname[128];
    char *basename;

    /* get the base file name for the deferred results */
    /*
     * Make it a directory under /tmp so at least we can set it to a
     * symbolic link in case /tmp doesn't have enough room.
     */
    basename = getenv("TPCC_RESULTS_FILE");
    if (basename == NULL)
        basename = "/tmp/TPCC_RESULTS_FILE";

    /* create the full file name */
    sprintf(fullname, "%s.%d", basename, id);

    /* open the file */
    unlink(fullname);
    rfile = fopen(fullname, "wb");
    if (rfile == NULL)
        syserror("Delivery server %d can't open file %s\n", id, fullname);

    /* allocate a larger buffer */
    }



results(t)
    delivery_trans *t;
    {
    if (fwrite(t, sizeof(*t), 1, rfile) != 1)
        syserror("Delivery server: Can't post results\n");
    }
```

```
results_close()
    {
    if (fclose(rfile) < 0)
        syserror("Delivery server can't close file\n");
    }
```

A.2 Server Stored Procedures

client/oracle/blocks/tkvcpnew.sql

```
-- New Order Anonymous block

  DECLARE
      idx                     BINARY_INTEGER;
      dummy_local             BINARY_INTEGER;
      not_serializable        EXCEPTION;
      PRAGMA EXCEPTION_INIT(not_serializable,-8177);
      deadlock                EXCEPTION;
      PRAGMA EXCEPTION_INIT(deadlock,-60);
      snapshot_too_old        EXCEPTION;
      PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
   PROCEDURE u1 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_01,
            DECODE (instr(i_data,'original'), 0, 'G',
                    DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
:brand_generic;

   END u1;

   PROCEDURE u2 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_02,
            DECODE (instr(i_data,'original'), 0, 'G',
                    DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                             :brand_generic;
   END u2;

   PROCEDURE u3 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
```

```
         WHERE i_id = :ol_i_id(idx)
         AND s_w_id = :ol_supply_w_id(idx)
         RETURNING i_price, i_name, s_quantity, s_dist_03,
           DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
          BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u3;

   PROCEDURE u4 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_04,
            DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u4;

   PROCEDURE u5 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_05,
            DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u5;

   PROCEDURE u6 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_06,
            DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u6;

   PROCEDURE u7 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_07,
           DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
          BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u7;

   PROCEDURE u8 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_08,
            DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u8;

   PROCEDURE u9 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_09,
            DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u9;

   PROCEDURE u10 IS
   BEGIN
         FORALL idx IN 1 .. :o_ol_cnt
          UPDATE stock_item
          SET s_order_cnt = s_order_cnt + 1,
          s_ytd = s_ytd + :ol_quantity(idx),
          s_remote_cnt = s_remote_cnt + :s_remote(idx),
          s_quantity = s_quantity - :ol_quantity(idx) +
                       DECODE(sign(s_quantity - :ol_quantity(idx) -
10),-1,91,0)
          WHERE i_id = :ol_i_id(idx)
          AND s_w_id = :ol_supply_w_id(idx)
          RETURNING i_price, i_name, s_quantity, s_dist_10,
            DECODE (instr(i_data,'original'), 0, 'G',
                     DECODE(instr(s_data,'original'), 0, 'G', 'B'))
           BULK COLLECT INTO :i_price, :i_name, :s_quantity, initnew.s_dist,
                           :brand_generic;
   END u10;

   PROCEDURE fix_items IS
     rows_lost                      BINARY_INTEGER;
     max_index                      BINARY_INTEGER;
     temp_index                     BINARY_INTEGER;
   BEGIN
        idx := 1;
        rows_lost := 0;
          max_index := sql%rowcount;

        WHILE (max_index != :o_ol_cnt) LOOP

            WHILE (idx <= sql%rowcount AND
                           sql%bulk_rowcount(idx + rows_lost) = 1)
          LOOP
              idx := idx + 1;
          END LOOP;
```

```
       temp_index := max_index;
       WHILE (temp_index >= idx + rows_lost) LOOP
            :i_price(temp_index + 1)       :=  :i_price(temp_index);
          :i_name(temp_index + 1)        :=  :i_name(temp_index);
            :s_quantity(temp_index + 1)    :=  :s_quantity(temp_index);
          initnew.s_dist(temp_index + 1) :=  initnew.s_dist(temp_index);
            :brand_generic(temp_index + 1) :=  :brand_generic(temp_index);
            temp_index := temp_index - 1;
       END LOOP;

        IF (idx + rows_lost <= :o_ol_cnt) THEN
           :i_price(idx + rows_lost)     :=   0;
           :i_name(idx + rows_lost)      :=   NULL;
         :s_quantity(idx + rows_lost)  :=   0;
         initnew.s_dist(idx + rows_lost) := NULL;
         :brand_generic(idx + rows_lost) := NULL;

         rows_lost := rows_lost + 1;
         max_index := max_index + 1;
          END IF;

     END LOOP;
END fix_items;

BEGIN
   LOOP BEGIN
      UPDATE district SET d_next_o_id = d_next_o_id + 1
       WHERE d_id = :d_id AND  d_w_id = :w_id
       RETURNING d_tax, d_next_o_id-1
       INTO :d_tax, :o_id;

      SELECT c_discount, c_last, c_credit, w_tax
       INTO :c_discount, :c_last, :c_credit , :w_tax
       FROM customer , warehouse
       WHERE c_id = :c_id AND c_d_id = :d_id AND c_w_id = :w_id
       AND  w_id = :w_id;


      INSERT INTO new_order (no_o_id, no_d_id, no_w_id)
       VALUES (:o_id, :d_id, :w_id);
       INSERT INTO orders  (o_id,o_d_id, o_w_id, o_c_id, o_entry_d,
                            o_carrier_id, o_ol_cnt, o_all_local)
       VALUES (:o_id, :d_id, :w_id, :c_id,
               :cr_date, 11, :o_ol_cnt, :o_all_local);

      dummy_local :=  :d_id;

      IF (dummy_local = 1) THEN u1; END IF;

      IF (dummy_local = 2) THEN u2; END IF;

      IF (dummy_local = 3) THEN u3; END IF;

      IF (dummy_local = 4) THEN u4; END IF;

      IF (dummy_local = 5) THEN u5; END IF;

      IF (dummy_local = 6) THEN u6; END IF;

      IF (dummy_local = 7) THEN u7; END IF;

      IF (dummy_local = 8) THEN u8; END IF;

      IF (dummy_local = 9) THEN u9; END IF;

      IF (dummy_local = 10) THEN u10; END IF;

      dummy_local := sql%rowcount;


      IF (dummy_local != :o_ol_cnt )  THEN fix_items; END IF;


     FOR idx IN 1 ..:o_ol_cnt LOOP
      :ol_amount(idx):=:ol_quantity(idx)*:i_price(idx);
     END LOOP;

     FORALL idx IN 1..:o_ol_cnt
       INSERT INTO order_line
          (ol_o_id, ol_d_id, ol_w_id, ol_number, ol_delivery_d, ol_i_id,
                ol_supply_w_id, ol_quantity,ol_amount,ol_dist_info)
       VALUES (:o_id, :d_id, :w_id, initnew.idx1arr(idx), initnew.nulldate,
               :ol_i_id(idx), :ol_supply_w_id(idx),
                 :ol_quantity(idx), :ol_amount(idx), initnew.s_dist(idx));

        IF (dummy_local != :o_ol_cnt) THEN
         :o_ol_cnt := dummy_local;
         ROLLBACK;
       END IF;

       EXIT;

       EXCEPTION
          WHEN not_serializable OR deadlock OR snapshot_too_old THEN
              ROLLBACK;
              :retry := :retry + 1;
        END;
     END LOOP;
  END;
```

client/oracle/blocks/payz.sql

```
DECLARE /* payz */
      not_serializable         EXCEPTION;
      PRAGMA EXCEPTION_INIT(not_serializable,-8177);
      deadlock                 EXCEPTION;
      PRAGMA EXCEPTION_INIT(deadlock,-60);
      snapshot_too_old         EXCEPTION;
      PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
   BEGIN
      LOOP BEGIN
         UPDATE warehouse
            SET w_ytd = w_ytd+:h_amount
            WHERE w_id = :w_id
            RETURNING w_name,
                      w_street_1, w_street_2, w_city, w_state, w_zip
              INTO initpay.ware_name,
                   :w_street_1, :w_street_2, :w_city, :w_state, :w_zip;


         SELECT rowid
         BULK COLLECT INTO initpay.row_id
         FROM customer
         WHERE c_d_id = :c_d_id AND c_w_id = :c_w_id AND c_last = :c_last
         ORDER BY c_last, c_d_id, c_w_id, c_first;

       initpay.c_num := sql%rowcount;
       initpay.cust_rowid := initpay.row_id((initpay.c_num+1) / 2);

        UPDATE customer
          SET c_balance = c_balance - :h_amount,
              c_ytd_payment = c_ytd_payment+ :h_amount,
              c_payment_cnt = c_payment_cnt+1
        WHERE rowid = initpay.cust_rowid
        RETURNING
                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
           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_data := ' ';
        IF :c_credit = 'BC' THEN
           UPDATE customer
             SET c_data = substr ((to_char (:c_id) || ' ' ||
                                   to_char (:c_d_id) || ' ' ||
                                   to_char (:c_w_id) || ' ' ||
                                   to_char (:d_id) || ' ' ||
                                   to_char (:w_id) || ' ' ||
                                   to_char (:h_amount/100, '9999.99') || ' | ')
```

```
                                || c_data, 1, 500)
            WHERE rowid = initpay.cust_rowid
            RETURNING substr(c_data,1, 200)
            INTO :c_data;

         END IF;

         UPDATE district
            SET d_ytd = d_ytd+:h_amount
            WHERE d_id = :d_id
              AND d_w_id = :w_id
            RETURNING  d_name, d_street_1, d_street_2, d_city,
                  d_state, d_zip
            INTO initpay.dist_name, :d_street_1, :d_street_2, :d_city,
                 :d_state, :d_zip;

            IF  SQL%NOTFOUND
              THEN
                    raise NO_DATA_FOUND;
            END IF;

         INSERT INTO history (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
                              h_amount, h_date, h_data)
             VALUES (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
             :cr_date, initpay.ware_name || '    ' || initpay.dist_name);

         EXIT;

         EXCEPTION
            WHEN not_serializable OR deadlock OR snapshot_too_old THEN
               ROLLBACK;
               :retry := :retry + 1;
         END;

      END LOOP;
   END;
```

client/oracle/blocks/paynz.sql

```
DECLARE /* paynz */
--      cust_rowid              ROWID;
--      dist_name               VARCHAR2(11);
--      ware_name               VARCHAR2(11);
      not_serializable         EXCEPTION;
      PRAGMA EXCEPTION_INIT(not_serializable,-8177);
      deadlock                 EXCEPTION;
      PRAGMA EXCEPTION_INIT(deadlock,-60);
      snapshot_too_old         EXCEPTION;
      PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
   BEGIN
      LOOP BEGIN
            UPDATE warehouse
               SET w_ytd = w_ytd + :h_amount
             WHERE w_id = :w_id
         RETURNING w_name, w_street_1, w_street_2, w_city, w_state, w_zip
              INTO initpay.ware_name, :w_street_1, :w_street_2, :w_city,
                   :w_state, :w_zip;

           UPDATE  customer
              SET  c_balance = c_balance - :h_amount,
                   c_ytd_payment = c_ytd_payment + :h_amount,
                   c_payment_cnt = c_payment_cnt+1
          WHERE    c_id = :c_id AND c_d_id = :c_d_id AND
                   c_w_id = :c_w_id
         RETURNING rowid, 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
              INTO initpay.cust_rowid,: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;

            IF SQL%NOTFOUND THEN
              raise NO_DATA_FOUND;
            END IF;


         IF :c_credit = 'BC' THEN
            UPDATE customer
                 SET c_data= substr ((to_char (:c_id) || ' ' ||
                                      to_char (:c_d_id) || ' ' ||
                                      to_char (:c_w_id) || ' ' ||
                                      to_char (:d_id) || ' ' ||
                                      to_char (:w_id) || ' ' ||
                                      to_char (:h_amount, '9999.99') || ' | ')
                                      || c_data, 1, 500)
             WHERE rowid = initpay.cust_rowid
         RETURNING substr(c_data,1, 200)
              INTO :c_data;

         END IF;

            UPDATE district
               SET d_ytd = d_ytd + :h_amount
             WHERE d_id = :d_id
               AND d_w_id = :w_id
         RETURNING d_name, d_street_1, d_street_2, d_city,d_state, d_zip
              INTO initpay.dist_name,:d_street_1,:d_street_2,:d_city,:d_state,
                   :d_zip;
          IF SQL%NOTFOUND THEN
            raise NO_DATA_FOUND;
          END IF;


         INSERT INTO history  (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
                               h_amount, h_date, h_data)
         VALUES
            (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
             :cr_date, initpay.ware_name || '    ' || initpay.dist_name);
         EXIT;

         EXCEPTION
            WHEN not_serializable OR deadlock OR snapshot_too_old THEN
               ROLLBACK;
               :retry := :retry + 1;
         END;

      END LOOP;
   END;
```

Appendix B Database Design

The source code for the process to define, create and populate the Oracle8i Enterprise Database Server v8.1.5 TPC-C database is included in this appendix.

B.1 Shell Scripts and Utility Shell Scripts

build/build.env

```
MULT=7488

# Env variables based on BUILD_HOME
BUILD_HOME=/BUILD
GEN_SQL=$BUILD_HOME/sql
LOAD_SCRIPTS=${BUILD_HOME}/scripts
TPCC_SQL=$BUILD_HOME/sql
BUILD_SQL=${BUILD_HOME}/sql
TPCC_LOADER=${BUILD_HOME}/loader
TPCC_UTILS=$BUILD_HOME/utils
OUTDIR=${BUILD_HOME}/outdir
LDIR=${BUILD_HOME}/data

TPCC_BLOCKS=${BUILD_HOME}/blocks

# Env variables based on BENCH_HOME
BENCH_HOME=$ORACLE_HOME/bench/tpc
TPCC_SOURCE=$BENCH_HOME/tpcc/source
TPCC_STORE=$BENCH_HOME/tpcc/stored_proc
#TPCC_BLOCKS=$BENCH_HOME/tpcc/blocks
TPCC_SCRIPTS=$BENCH_HOME/tpcc/scripts
AUDIT_SQL=$BENCH_HOME/tpcc/audit/sql

BENCH_GEN=$ORACLE_HOME/bench/gen

PATH=${TPCC_SOURCE}:${TPCC_UTILS}:${LOAD_SCRIPTS}:${PATH}
export PATH
```

build/benchsetup.sh

```
#!/bin/ksh -v

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#==================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA       |
#                    OPEN SYSTEMS PERFORMANCE GROUP                |
#                         All Rights Reserved                      |
#==================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#==================================================================
#

. build.env

STEP=0
START=0
END=0
CONTINUE=1
PROGNAME=$0


function usage {
   set -
   echo ""
  echo "Usage:        $PROGNAME [<start> <stop>] [<start>] [-step <stepno>]"
   echo "       [<start> <stop>]        - allows user to run a specified"
   echo "                                 range of steps."
   echo "       [<start>]               - runs from step number <start> till"
   echo "                                 the end of the script."
   echo "       [-step <stepno>]        - runs only step number <stepno> and"
   echo "                                 then stops."
   echo ""
   echo "               STEP    FUNCTION"
   echo "------------------------------------------------------------------"
   echo "               0       Create DB."
   echo "               1       Create user tpcc."
   echo "               2       Create warehouse table."
   echo "               3       Create district table."
   echo "               4       Create history table."
   echo "               5       Create Orders table."
   echo "               6       Create New-order table."
   echo "               7       Create Orderline table."
   echo "               8       Create Item table."
   echo "               9       Create Customer table."
   echo "               10      Create Stock table."
   echo "               11      Create rollback segments."
   echo "               12      Load New-order."
   echo "               13      Load History."
   echo "               14      Load Order/Orderline."
   echo "               15      Load Warehouse."
   echo "               16      Load District."
   echo "               17      Load Item."
   echo "               18      Load Customer."
   echo "               19      Load Stock"
   echo "               20      Alter temp space."
   echo "               21      Create Warehouse index."
   echo "               22      Create District index."
   echo "               23      Create Item index."
   echo "               24      Create Customer index."
   echo "               25      Create Customer2 index."
   echo "               26      Create Stock index."
   echo "               27      Create Orders index."
   echo "               28      Create Orders2 index."
   echo "               29      Create New-order index."
   echo "               30      Create Orderline index."
   echo "               31      Re-alter temp space."
   echo "               32      Analyze."
   echo "               33      Create TPC-C reports tables."
   echo "               34      Create stored procs."
   echo "               35      Space rpts / etc."
   echo "               36      Alter extents and Lock tables."
   echo "               37      Run catalog scripts."
   echo "               38      Shutdown database."
   echo "------------------------------------------------------------------"

   exit 1;
}

function runnable {
  echo `date` "step $STEP"
  if [ -a "./stop" ]
  then
    exit 1;
  fi
  if [ $STEP -ge $START ]
  then
    if [ $STEP -le $END ]
    then
        STEP=`expr $STEP + 1`;
        return 0;
    else
        if [ $CONTINUE = 0 ]
        then
           STEP=`expr $STEP + 1`;
```

```
          return 0;
        fi
    fi
  fi
  STEP=`expr $STEP + 1`;
  return 1;
}

case $# in
        0)      usage
                CONTINUE=0
                ;;
        1)      case $1 in
                        -h)     usage
                                ;;
                        *)      START=$1
                                CONTINUE=0
                                ;;
                esac
                ;;
        2)      case $1 in
                        -step)  shift
                                START=$1
                                END=$1
                                CONTINUE=1
                                ;;
                        *)      START=$1
                                shift
                                END=$1
                                CONTINUE=1
                                ;;
                esac
                ;;
        *)      usage
                ;;
esac

if [ ! -d $BUILD_HOME ]
then
   mkdir $BUILD_HOME
fi

if [ ! -d $LOAD_SCRIPTS ]
then
   mkdir $LOAD_SCRIPTS
fi

if [ ! -d $OUTDIR ]
then
   mkdir $OUTDIR
fi

if [ ! -d $LDIR ]
then
   mkdir $LDIR
fi


if runnable
then
benchdb.sh > ${OUTDIR}/benchdb.out 2>&1
  echo "Switching Logs ..."
  ${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1
fi


if runnable
then
${LOAD_SCRIPTS}/create_user.sh > ${OUTDIR}/create_user.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_ware.sh > ${OUTDIR}/create_ware.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_dist.sh > ${OUTDIR}/create_dist.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_hist.sh > ${OUTDIR}/create_hist.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_ordr.sh > ${OUTDIR}/create_ordr.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_nord.sh > ${OUTDIR}/create_nord.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_ordl.sh > ${OUTDIR}/create_ordl.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_item.sh > ${OUTDIR}/create_item.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_cust.sh > ${OUTDIR}/create_cust.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/create_stok.sh > ${OUTDIR}/create_stok.out 2>&1 &
fi

wait

if runnable
then
(
${LOAD_SCRIPTS}/tpcc_rol.sh > ${OUTDIR}/tpcc_rol.out 2>&1
sqlplus sys/change_on_install @alterroll >> ${OUTDIR}/tpcc_rol.out 2>&1
) &
fi

echo "Switching Logs ..."
${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1

if runnable
then
${LOAD_SCRIPTS}/load_nord.sh > ${OUTDIR}/load_nord.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/load_hist.sh > ${OUTDIR}/load_hist.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/load_ordr.sh > ${OUTDIR}/load_ordr.out 2>&1
fi

wait

echo "Switching Logs ..."
${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1

if runnable
then
${LOAD_SCRIPTS}/load_ware.sh > ${OUTDIR}/load_ware.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/load_dist.sh > ${OUTDIR}/load_dist.out 2>&1
fi
```

```
if runnable
then
${LOAD_SCRIPTS}/load_item.sh > ${OUTDIR}/load_item.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/load_cust.sh > ${OUTDIR}/load_cust.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/load_stok.sh > ${OUTDIR}/load_stok.out 2>&1 &
fi

wait

echo "Switching Logs ..."
${TPCC_UTILS}/switchlog.sh >> ${OUTDIR}/switchlog.out 2>&1


if runnable
then
${LOAD_SCRIPTS}/alter_temp.sh > ${OUTDIR}/alter_temp.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_iware.sh > ${OUTDIR}/create_iware.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_idist.sh > ${OUTDIR}/create_idist.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_iitem.sh > ${OUTDIR}/create_iitem.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/create_icust.sh > ${OUTDIR}/create_icust.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/create_icust2.sh > ${OUTDIR}/create_icust2.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/create_istok.sh > ${OUTDIR}/create_istok.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/create_iordr.sh > ${OUTDIR}/create_iordr.out 2>&1 &
fi

if runnable
then
${LOAD_SCRIPTS}/create_iordr2.sh > ${OUTDIR}/create_iordr2.out 2>&1 &
fi

if runnable
then
echo "create_inord.sh not called since IOT"
## ${LOAD_SCRIPTS}/create_inord.sh > ${OUTDIR}/create_inord.out 2>&1
fi

if runnable
then
echo "create_iordl.sh not called since IOT"
##${LOAD_SCRIPTS}/create_iordl.sh > ${OUTDIR}/create_iordl.out 2>&1
fi

wait

if runnable
then
${LOAD_SCRIPTS}/realter_temp.sh > ${OUTDIR}/realter_temp.out 2>&1
fi


if runnable
then
sqlplus tpcc/tpcc @$TPCC_SQL/tpcc_ana > ${OUTDIR}/tpcc_ana.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/tpcc_reports.sh > ${OUTDIR}/tpcc_reports.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/tpcc_stored_proc.sh > ${OUTDIR}/tpcc_stored_prod.out 2>&1
$TPCC_UTILS/create_cache_views.sh > ${OUTDIR}/create_cache_views.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/tpcc_misc.sh > ${OUTDIR}/tpcc_misc.out 2>&1
fi

if runnable
then
echo "alter.sh not called since local uniform storage management"
## ${LOAD_SCRIPTS}/alter.sh > ${OUTDIR}/alter.out 2>&1
${TPCC_UTILS}/dml.sh > ${OUTDIR}/dml.out 2>&1
${LOAD_SCRIPTS}/cust_recycle.sh > ${OUTDIR}/cust_recycle.out 2>&1
${LOAD_SCRIPTS}/hist_recycle.sh > ${OUTDIR}/hist_recycle.out 2>&1
${LOAD_SCRIPTS}/stock_keep.sh > ${OUTDIR}/stock_keep.out 2>&1
fi

if runnable
then
${LOAD_SCRIPTS}/cat.sh > ${OUTDIR}/cat.out 2>&1
fi

if runnable
then
svrmgrl <<!
   connect internal;
   alter system switch logfile;
   alter system switch logfile;
   shutdown;
   exit;
!
fi
```

build/benchdb.sh

```
#!/bin/ksh

#
# benchdb.sh 8030100 98/6/1 16:17 vmakhija
# Copyr (c) 1998 Oracle
#

#
#===================================================================+
#        Copyright (c) 1997  Oracle Corp, Redwood Shores, CA        |
#                    OPEN SYSTEMS PERFORMANCE GROUP                 |
#                         All Rights Reserved                       |
#===================================================================+
# FILENAME
#      benchdb.sh
# DESCRIPTION
#      Usage: benchdb.sh [options]
#             -n           do not create new tpcc database
#             -c           do not run catalog scripts
```

```
#            -t          do not create temp tablespace
#================================================================
#

. build.env
TPCC_ADMIN=$BUILD_HOME/admin

while [ "$#" != "0" ]
do
   case $1 in
       -n) shift
           NO_CREATE="y"
           ;;
       -c) shift
           NO_CAT="y"
           ;;
       -t) shift
           NO_TEMP="y"
           ;;
       -s) shift
           NO_TABLESPACES="y"
           ;;
        *) echo "Bad arg: $1"
           exit 1;
           ;;
   esac
done


#
# Create database if NO_CREATE unset
#

if [ "$NO_CREATE" = "" ]
then
svrmgrl <<!
        set echo on
        connect internal
        startup  pfile=$TPCC_ADMIN/p_create.ora  nomount
        create database tpcc controlfile reuse maxdatafiles 400
            datafile '?/dbs/${ORACLE_SID}_disks/sys1' size 801M reuse
            logfile  '?/dbs/${ORACLE_SID}_disks/tmplog1' size 3200M reuse,
                     '?/dbs/${ORACLE_SID}_disks/tmplog2' size 3200M reuse;
        exit
!


#
# Create more rollback segments
#

svrmgrl <<!
   connect system/manager
   create rollback segment s1  storage (initial 200k minextents 2 next 200k);
   create rollback segment s2  storage (initial 200k minextents 2 next 200k);
   create rollback segment s3  storage (initial 200k minextents 2 next 200k);
   create rollback segment s4  storage (initial 200k minextents 2 next 200k);
   create rollback segment s5  storage (initial 200k minextents 2 next 200k);
   create rollback segment s6  storage (initial 200k minextents 2 next 200k);
   create rollback segment s7  storage (initial 200k minextents 2 next 200k);
   create rollback segment s8  storage (initial 200k minextents 2 next 200k);
   create rollback segment s9  storage (initial 200k minextents 2 next 200k);
   create rollback segment s10 storage (initial 200k minextents 2 next 200k);
   create rollback segment s11 storage (initial 200k minextents 2 next 200k);
   create rollback segment s12 storage (initial 200k minextents 2 next 200k);
   create rollback segment s13 storage (initial 200k minextents 2 next 200k);
   create rollback segment s14 storage (initial 200k minextents 2 next 200k);
   create rollback segment s15 storage (initial 200k minextents 2 next 200k);
   create rollback segment s16 storage (initial 200k minextents 2 next 200k);
   create rollback segment s17 storage (initial 200k minextents 2 next 200k);
   create rollback segment s18 storage (initial 200k minextents 2 next 200k);
   create rollback segment s19 storage (initial 200k minextents 2 next 200k);
   create rollback segment s20 storage (initial 200k minextents 2 next 200k);
   create rollback segment s21 storage (initial 200k minextents 2 next 200k);
   create rollback segment s22 storage (initial 200k minextents 2 next 200k);
   create rollback segment s23 storage (initial 200k minextents 2 next 200k);
   create rollback segment s24 storage (initial 200k minextents 2 next 200k);
   create rollback segment s25 storage (initial 200k minextents 2 next 200k);
   create rollback segment s26 storage (initial 200k minextents 2 next 200k);
   create rollback segment s27 storage (initial 200k minextents 2 next 200k);
   create rollback segment s28 storage (initial 200k minextents 2 next 200k);
   create rollback segment s29 storage (initial 200k minextents 2 next 200k);
   create rollback segment s30 storage (initial 200k minextents 2 next 200k);
   disconnect;
   connect internal;
   shutdown;
   exit;
!
fi


#
#  Startup database with params file that includes new rollback segments
#

if [ "$NO_TEMP" = "" ]
then
svrmgrl <<!
        connect internal
        startup pfile=$TPCC_ADMIN/p_build.ora;
        connect system/manager
        create tablespace temp datafile
           '?/dbs/${ORACLE_SID}_disks/temp01'  size  7651M reuse;
        exit;
!

fi


#
#       Add tablespaces in parallel
#

if [ "$NO_TABLESPACES" = "" ]
then
addts.sh stok   \?/dbs/${ORACLE_SID}_disks/stock01  3051M   50M &
addts.sh cust   \?/dbs/${ORACLE_SID}_disks/cust01   2289M   44M &
addts.sh ordl   \?/dbs/${ORACLE_SID}_disks/ordl01   7651M  765M &
addts.sh nord   \?/dbs/${ORACLE_SID}_disks/nord01   2501M  250M &
addts.sh ordr   \?/dbs/${ORACLE_SID}_disks/ordr01   7251M  725M &
addts.sh hist   \?/dbs/${ORACLE_SID}_disks/hist01   5001M 1000M &

addts.sh istk   \?/dbs/${ORACLE_SID}_disks/istk01   7501M  150M &
addts.sh icust1 \?/dbs/${ORACLE_SID}_disks/icust101 8001M  100M &
addts.sh icust2 \?/dbs/${ORACLE_SID}_disks/icust201 8001M  100M &
addts.sh iord1  \?/dbs/${ORACLE_SID}_disks/iord101  5501M  250M &
addts.sh iord2  \?/dbs/${ORACLE_SID}_disks/iord201  8001M  400M &

# Rollback segments
addroll.sh \?/dbs/${ORACLE_SID}_disks/roll01 401M &

wait

fi

#
#         Add datafiles to tablespaces in parallel
#

addft.sh roll   \?/dbs/${ORACLE_SID}_disks/roll    400M  1 &
addft.sh stok   \?/dbs/${ORACLE_SID}_disks/stock  3051M 95 &
addft.sh cust   \?/dbs/${ORACLE_SID}_disks/cust   2289M 95 &
wait
addft.sh hist   \?/dbs/${ORACLE_SID}_disks/hist   5001M  3 &
addft.sh istk   \?/dbs/${ORACLE_SID}_disks/istk   7501M  3 &
addft.sh icust2 \?/dbs/${ORACLE_SID}_disks/icust2 8001M  1 &

addft.sh iord1  \?/dbs/${ORACLE_SID}_disks/iord1  5501M  1 &
addft.sh iord2  \?/dbs/${ORACLE_SID}_disks/iord2  8001M  1 &
addft.sh ordr   \?/dbs/${ORACLE_SID}_disks/ordr   7251M  1 &
addft.sh temp   \?/dbs/${ORACLE_SID}_disks/temp   7651M 11 &
addft.sh ordl   \?/dbs/${ORACLE_SID}_disks/ordl   7651M 34 &

wait

svrmgrl <<!
        set echo off;
        connect internal;
```

```
        alter tablespace nord logging;
          alter tablespace ordl logging;
        exit;
!

#
# run catalog if NO_CAT unset
#

if [ "$NO_CAT" = "" ]
then
svrmgrl <<!
        set echo off;
        connect sys/change_on_install;
        @?/rdbms/admin/catalog;
        @?/rdbms/admin/catproc;
        @?/rdbms/admin/catparr;
        exit;
!
fi
```

build/add_97.sh

```
. build.env

addfile.sh stok \?/dbs/${ORACLE_SID}_disks/stock97 7500M &
addfile.sh stok \?/dbs/${ORACLE_SID}_disks/stock98 7500M &
addfile.sh cust \?/dbs/${ORACLE_SID}_disks/cust97 6000M &
addfile.sh cust \?/dbs/${ORACLE_SID}_disks/cust98 6000M &
wait
```

build/scripts/addfile.sh

```
#
# $Header: addfile.sh 7030100.1 96/05/02 10:30:04 plai Generic<base> $ Copyr (c)
1995 Oracle
#

#=================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#     addfile.sh
# DESCRIPTION
#     Add datafile to a tablespace.
# USAGE
#     addfile.sh <tablespace> <data file> <size>
#================================================================*/

FILE=`basename $2`

if [ -d ./outdir ]
then
   echo `date` > ./outdir/${FILE}.addf
fi

svrmgrl <<!
   connect internal
   set echo on
   alter tablespace $1 add datafile '$2' size $3 reuse;
   exit;
!

if [ -d ./outdir ]
then
   echo `date` >> ./outdir/${FILE}.addf
fi
```

build/scripts/addft.sh

```
#!/bin/ksh

. build.env

TNAME=$1
FNAME=$2
SIZE=$3
NFILE=`expr $4 + 1`

I=2
while [ $I -le $NFILE ]
do
N=`printf "%02d" $I`
addfile.sh $TNAME ${FNAME}$N $SIZE &
I=`expr $I + 1`
done
wait
```

build/scripts/addroll.sh

```
#!/bin/ksh
#=================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#     addts.sh
# DESCRIPTION
#     Add tablespace to database.
# USAGE
#     addts.sh <tablespace> <data file> <size>
#================================================================*/
echo 'ORACLE_HOME=' $ORACLE_HOME
echo 'ORACLE_SID=' $ORACLE_SID

FILE=`basename $1`

if [ -d ./outdir ]
then
   echo `date` > ./outdir/${FILE}.addts
fi

svrmgrl <<!
   connect internal
   create tablespace roll datafile '$1' size $2 reuse extent management local
uniform size 40K nologging ;

   exit;
!

if [ -d ./outdir ]
then
   echo `date` >> ./outdir/${FILE}.addts
fi
```

build/scripts/addts.sh

```
#!/bin/ksh
#=================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#     addts.sh
```

```
# DESCRIPTION
#     Add tablespace to database.
# USAGE
#     addts.sh <tablespace> <data file> <size>
#=====================================================================*/

FILE=`basename $2`

if [ -d ./outdir ]
then
   echo `date` > ./outdir/${FILE}.addts
fi

svrmgrl <<!
   connect internal
   set echo on
   create tablespace $1 datafile '$2' size $3 reuse extent management local uniform
size $4 nologging ;

   exit;
!

if [ -d ./outdir ]
then
   echo `date` >> ./outdir/${FILE}.addts
fi
```

build/scripts/alter_temp.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                        All Rights Reserved                          |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=====================================================================
#

. build.env


sqlplus system/manager <<!
   alter user tpcc temporary tablespace temp;
   quit;
!

svrmgrl <<!
   connect internal
   alter tablespace temp
      default storage (initial 33M next 33M pctincrease 0);
   exit;
!
```

buildscripts/cat.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                        All Rights Reserved                          |
#=====================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=====================================================================
#

. build.env

svrmgrl <<!
   set echo off;
   connect sys/change_on_install;
   @?/rdbms/admin/catparr;
   exit;
!
```

build/scripts/create_cust/sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                        All Rights Reserved                          |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

. build.env

sqlplus tpcc/tpcc @cust
```

build/scripts/create_dist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=====================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA          |
#                    OPEN SYSTEMS PERFORMANCE GROUP                   |
#                        All Rights Reserved                          |
#=====================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=====================================================================
#

. build.env

sqlplus tpcc/tpcc @dist
```

build/ scripts/create_hist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @hist
```

build/ scripts/create_icust.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @icust
```

build/ scripts/create_icust2.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @icust2
```

build/scripts/create_dist.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @dist
```

build/create_iitem.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @iitem
```

build/ scripts/create_iordr.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
```

```
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=============================================================
#

. build.env

sqlplus tpcc/tpcc @iordr
```

build/ scripts/create_iordr2.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=============================================================
#

. build.env

sqlplus tpcc/tpcc @iordr2
```

build/ scripts/create_istok.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=============================================================
#

. build.env

sqlplus tpcc/tpcc @istok
```

build/scripts/create_item.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=============================================================
#

. build.env

sqlplus tpcc/tpcc @item
```

build/scripts/create_iware.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=============================================================
#

. build.env

sqlplus tpcc/tpcc @iware
```

build/ scripts/create_nord.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=============================================================
#

. build.env

sqlplus tpcc/tpcc @nord
```

build/ scripts/create_ordl.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @ordl
```

build/ scripts/create_ordr.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @ordr
```

build/ scripts/create_stok.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

sqlplus tpcc/tpcc @stok
```

build/scripts/create_user.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env

svrmgrl <<!
rem
rem ===================================================================+
rem         Copyright (c) 1997  Oracle Corp, Redwood Shores, CA        |
rem                     OPEN SYSTEMS PERFORMANCE GROUP                |
rem                        All Rights Reserved                        |
rem ===================================================================+
rem  FILENAME
rem       tpcc_user.sql
rem  DESCRIPTION
rem       Create user for TPC-C database.
rem ====================================================================
rem

rem
rem Create TPCC userid and connect to it.
rem
        connect internal;
        grant connect,resource,unlimited tablespace to tpcc identified by tpcc;
        alter user tpcc temporary tablespace temp;
        connect tpcc/tpcc;
        exit;
!
```

build/scripts/create_ware.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#===================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA        |
#                     OPEN SYSTEMS PERFORMANCE GROUP                |
#                          All Rights Reserved                      |
#===================================================================+
# FILENAME
#      create_<obj>.sh
# DESCRIPTION
#      Usage: create_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#====================================================================
#

. build.env
```

```
sqlplus tpcc/tpcc @ware
```

build/scripts/cust_recycle.sh

```
sqlplus tpcc/tpcc <<!
alter table customer enable table lock;
alter cluster ccluster storage ( buffer_pool recycle );
alter table customer disable table lock;
exit;
!
```

build/scripts/hist_recycle.sh

```
sqlplus tpcc/tpcc <<!
alter table history enable table lock;
alter table history storage ( buffer_pool recycle );
alter table history disable table lock;
exit;
!
```

build/scripts/load_cust.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

#
#  Load customer table (in parallel with loading stock table)
#

I=1
while [ $I -le 104 ]
do
   rm -f ${LDIR}/cust${I}.dat
   mknod ${LDIR}/cust${I}.dat p
   I=`expr $I + 1`
done

I=1
SW=1
EW=72
INC=72
while [ $I -le 104 ]
do
   tpccload -M $MULT -c -b $SW -e $EW -g > ${LDIR}/cust${I}.dat \
   2> ${OUTDIR}/cust${I}.out &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 10

I=1
while [ $I -le 104 ]
do
          sqlldr tpcc/tpcc control=$TPCC_LOADER/cust.ctl \
      log=${OUTDIR}/cust${I}.log \
      bad=${OUTDIR}/cust${I}.bad data=${LDIR}/cust${I}.dat \
      discard=${OUTDIR}/cust${I}.dsc \
      > /dev/null &
          I=`expr $I + 1`
done

wait

I=1
while [ $I -le 104 ]
do
   rm -f ${LDIR}/cust${I}.dat
   I=`expr $I + 1`
done
```

build/scripts/load_dist.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

#tpccload -M $MULT -d

#
#  Load district table
#

rm -f ${LDIR}/dist.dat
mknod ${LDIR}/dist.dat p

tpccload -M $MULT -d -g > ${LDIR}/dist.dat 2> \
      ${OUTDIR}/dist.out &

sleep 2
sqlldr tpcc/tpcc control=$TPCC_LOADER/dist.ctl \
      log=${OUTDIR}/dist.log \
      bad=${OUTDIR}/dist.bad data=${LDIR}/dist.dat \
      discard=${OUTDIR}/dist.dsc \
      > /dev/null &

wait

rm -f ${LDIR}/dist.dat
```

build/scripts/load_hist.sh

```
#!/bin/ksh
```

```
#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: load_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

. build.env


#
#  Load history table
#

I=1
while [ $I -le 24 ]
do
   rm -f ${LDIR}/hist${I}.dat
   mknod ${LDIR}/hist${I}.dat p
   I=`expr $I + 1`
done

I=1
SW=1
EW=312
INC=312
while [ $I -le 24 ]
do
   tpccload -M $MULT -h -g -b $SW -e $EW > ${LDIR}/hist${I}.dat 2> \
      ${OUTDIR}/hist${I}.out &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 10

I=1
while [ $I -le 24 ]
do
   sqlldr tpcc/tpcc control=$TPCC_LOADER/hist.ctl \
          log=${OUTDIR}/hist${I}.log \
          bad=${OUTDIR}/hist${I}.bad \
          data=${LDIR}/hist${I}.dat \
          discard=${OUTDIR}/hist${I}.dsc \
      > /dev/null &
   I=`expr $I + 1`
done

wait

I=1
while [ $I -le 24 ]
do
   rm -f ${LDIR}/hist${I}.dat
   I=`expr $I + 1`
done
```

build/scripts/load_item.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

#tpccload -M $MULT -i

rm -f ${LDIR}/item.dat
mknod ${LDIR}/item.dat p

tpccload -M $MULT -i -g > ${LDIR}/item.dat 2> \
      ${OUTDIR}/item.out &

sleep 2
sqlldr tpcc/tpcc control=$TPCC_LOADER/item.ctl \
      log=${OUTDIR}/item.log \
      bad=${OUTDIR}/item.bad data=${LDIR}/item.dat \
      discard=${OUTDIR}/item.dsc \
      > /dev/null &

wait

rm -f ${LDIR}/item.dat
```

build/scripts/load_nord.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: load_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

. build.env

#
#   Load new-order table
#

I=1
while [ $I -le 24 ]
do
   rm -f ${LDIR}/neword${I}.dat
   mknod ${LDIR}/neword${I}.dat p
   I=`expr $I + 1`
done

I=1
SW=1
EW=312
INC=312
while [ $I -le 24 ]
do
  tpccload -M $MULT -n -b $SW -e $EW -g > ${LDIR}/neword${I}.dat 2> \
```

```
      ${OUTDIR}/neword${I}.out &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done

sleep 10

I=1
while [ $I -le 24 ]
do
   sqlldr tpcc/tpcc control=$TPCC_LOADER/neword.ctl \
        log=${OUTDIR}/neword${I}.log \
        bad=${OUTDIR}/neword${I}.bad data=${LDIR}/neword${I}.dat \
        discard=${OUTDIR}/neword${I}.dsc \
      > /dev/null &
   I=`expr $I + 1`
done

wait

I=1
while [ $I -le 24 ]
do
   rm -f ${LDIR}/neword${I}.dat
   I=`expr $I + 1`
done
```

build/scripts/load_ordr.sh

```
#!/bin/ksh

#
# load_<obj>.sh 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle Corp.
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      pload.sh
# DESCRIPTION
#      Usage: load_<obj>.sh [options]
#             -mu <multiplier>     (# of warehouses)
#=================================================================
#

. build.env

#
#  Load order and order-line table
#

I=1
while [ $I -le 78 ]
do
   rm -f ${LDIR}/order${I}.dat
   rm -f ${LDIR}/ordline${I}.dat
   mknod ${LDIR}/order${I}.dat p
   mknod ${LDIR}/ordline${I}.dat p
   I=`expr $I + 1`
done

I=1
SW=1
EW=96
INC=96
while [ $I -le 78 ]
do
   tpccload -M $MULT -o ${LDIR}/ordline${I}.dat -g -b $SW -e $EW > \
      ${LDIR}/order${I}.dat 2> ${OUTDIR}/order${I}.out &
   I=`expr $I + 1`
   SW=`expr $SW + $INC`
   EW=`expr $EW + $INC`
done


sleep 10

I=1
while [ $I -le 78 ]
do
   sqlldr tpcc/tpcc control=$TPCC_LOADER/order.ctl \
          log=${OUTDIR}/order${I}.log \
          bad=${OUTDIR}/order${I}.bad data=${LDIR}/order${I}.dat \
          discard=${OUTDIR}/order${I}.dsc \
        > /dev/null &
   sqlldr tpcc/tpcc control=$TPCC_LOADER/ordline.ctl \
          log=${OUTDIR}/ordline${I}.log \
          bad=${OUTDIR}/ordline${I}.bad data=${LDIR}/ordline${I}.dat \
          discard=${OUTDIR}/ordline${I}.dsc \
        > /dev/null &
   I=`expr $I + 1`
done

wait

I=1
while [ $I -le 78 ]
do
   rm -f ${LDIR}/order${I}.dat
   rm -f ${LDIR}/ordline${I}.dat
   I=`expr $I + 1`
done
```

build/scripts/load_stok.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

#
#  Load stock table (in parallel with loading customer table)
#

I=1
while [ $I -le 250 ]
do
   rm -f ${LDIR}/stok${I}.dat
   mknod ${LDIR}/stok${I}.dat p
   I=`expr $I + 1`
done

I=1
SI=1
EI=400
INC=400
while [ $I -le 250 ]
do
   tpccload -M $MULT -S -j $SI -k $EI -g \
   > ${LDIR}/stok${I}.dat  2> ${OUTDIR}/stk${I}.out  &
   I=`expr $I + 1`
   SI=`expr $SI + $INC`
   EI=`expr $EI + $INC`
```

```
done

sleep 10

I=1
while [ $I -le 250 ]
do
          sqlldr tpcc/tpcc control=$TPCC_LOADER/stock.ctl \
      log=${OUTDIR}/stok${I}.log \
      bad=${OUTDIR}/stok${I}.bad data=${LDIR}/stok${I}.dat \
      discard=${OUTDIR}/stok${I}.dsc \
      > /dev/null &
          I=`expr $I + 1`
done

wait

I=1
while [ $I -le 250 ]
do
   rm -f ${LDIR}/stok${I}.dat
   I=`expr $I + 1`
done
```

build/scripts/load_ware.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

#tpccload -M $MULT -w

rm -f ${LDIR}/ware.dat
mknod ${LDIR}/ware.dat p

tpccload -M $MULT -w -g > ${LDIR}/ware.dat 2> \
      ${OUTDIR}/ware.out &

sleep 2

sqlldr tpcc/tpcc control=$TPCC_LOADER/ware.ctl \
      log=${OUTDIR}/ware.log \
      bad=${OUTDIR}/ware.bad data=${LDIR}/ware.dat \
      discard=${OUTDIR}/ware.dsc \
      > /dev/null &

wait

rm -f ${LDIR}/ware.dat
```

build/scripts/realter_temp.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

svrmgrl <<!
   connect internal
   alter tablespace temp
     default storage (initial 20K next 20K pctincrease 50);
   exit;
!
```

build/scripts/recreate_rol.sh.3

```
#!/bin/ksh
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#      tpccrol.sh
# DESCRIPTION
#      Script file for creating the roll;back segments.
#=================================================================
#

svrmgrl <<!
   connect internal;
   host date;
alter rollback segment S1 online;
alter rollback segment S2 online;
alter rollback segment S3 online;
alter rollback segment S4 online;
alter rollback segment S5 online;
alter rollback segment S6 online;
alter rollback segment S7 online;
alter rollback segment S8 online;
alter rollback segment S9 online;
alter rollback segment S10 online;
alter rollback segment S11 online;
alter rollback segment S12 online;
alter rollback segment S13 online;
alter rollback segment S14 online;
alter rollback segment S15 online;
alter rollback segment S16 online;
alter rollback segment S17 online;
alter rollback segment S18 online;
alter rollback segment S19 online;
alter rollback segment S20 online;
alter rollback segment S21 online;
alter rollback segment S22 online;
alter rollback segment S23 online;
alter rollback segment S24 online;
alter rollback segment S25 online;
alter rollback segment S26 online;
alter rollback segment S27 online;
alter rollback segment S28 online;
alter rollback segment S29 online;
alter rollback segment S30 online;

alter rollback segment T1 offline;
alter rollback segment T2 offline;
```

```
alter rollback segment T3 offline;
alter rollback segment T4 offline;
alter rollback segment T5 offline;
alter rollback segment T6 offline;
alter rollback segment T7 offline;
alter rollback segment T8 offline;
alter rollback segment T9 offline;
alter rollback segment T10 offline;
alter rollback segment T11 offline;
alter rollback segment T12 offline;
alter rollback segment T13 offline;
alter rollback segment T14 offline;
alter rollback segment T15 offline;
alter rollback segment T16 offline;
alter rollback segment T17 offline;
alter rollback segment T18 offline;
alter rollback segment T19 offline;
alter rollback segment T20 offline;
alter rollback segment T21 offline;
alter rollback segment T22 offline;
alter rollback segment T23 offline;
alter rollback segment T24 offline;
alter rollback segment T25 offline;
alter rollback segment T26 offline;
alter rollback segment T27 offline;
alter rollback segment T28 offline;
alter rollback segment T29 offline;
alter rollback segment T30 offline;
alter rollback segment T31 offline;
alter rollback segment T32 offline;
alter rollback segment T33 offline;
alter rollback segment T34 offline;
alter rollback segment T35 offline;
alter rollback segment T36 offline;
alter rollback segment T37 offline;
alter rollback segment T38 offline;
alter rollback segment T39 offline;
alter rollback segment T40 offline;
alter rollback segment T41 offline;
alter rollback segment T42 offline;
alter rollback segment T43 offline;
alter rollback segment T44 offline;
alter rollback segment T45 offline;
alter rollback segment T46 offline;
alter rollback segment T47 offline;
alter rollback segment T48 offline;
alter rollback segment T49 offline;
alter rollback segment T50 offline;
alter rollback segment T51 offline;
alter rollback segment T52 offline;
alter rollback segment T53 offline;
alter rollback segment T54 offline;
alter rollback segment T55 offline;
alter rollback segment T56 offline;
alter rollback segment T57 offline;
alter rollback segment T58 offline;
alter rollback segment T59 offline;
alter rollback segment T60 offline;
alter rollback segment T61 offline;
alter rollback segment T62 offline;
alter rollback segment T63 offline;
alter rollback segment T64 offline;
alter rollback segment T65 offline;
alter rollback segment T66 offline;
alter rollback segment T67 offline;
alter rollback segment T68 offline;
alter rollback segment T69 offline;
alter rollback segment T70 offline;
alter rollback segment T71 offline;
alter rollback segment T72 offline;
alter rollback segment T73 offline;
alter rollback segment T74 offline;
alter rollback segment T75 offline;
alter rollback segment T76 offline;
alter rollback segment T77 offline;
alter rollback segment T78 offline;
alter rollback segment T79 offline;
alter rollback segment T80 offline;
alter rollback segment T81 offline;
alter rollback segment T82 offline;
alter rollback segment T83 offline;
alter rollback segment T84 offline;
alter rollback segment T85 offline;
alter rollback segment T86 offline;
alter rollback segment T87 offline;
alter rollback segment T88 offline;
alter rollback segment T89 offline;
alter rollback segment T90 offline;
alter rollback segment T91 offline;
alter rollback segment T92 offline;
alter rollback segment T93 offline;
alter rollback segment T94 offline;
alter rollback segment T95 offline;
alter rollback segment T96 offline;
alter rollback segment T97 offline;
alter rollback segment T98 offline;
alter rollback segment T99 offline;
alter rollback segment T100 offline;
alter rollback segment T101 offline;
alter rollback segment T102 offline;
alter rollback segment T103 offline;
alter rollback segment T104 offline;
alter rollback segment T105 offline;
alter rollback segment T106 offline;
alter rollback segment T107 offline;
alter rollback segment T108 offline;
alter rollback segment T109 offline;
alter rollback segment T110 offline;
alter rollback segment T111 offline;
alter rollback segment T112 offline;
alter rollback segment T113 offline;
alter rollback segment T114 offline;
alter rollback segment T115 offline;
alter rollback segment T116 offline;
alter rollback segment T117 offline;
alter rollback segment T118 offline;
alter rollback segment T119 offline;
alter rollback segment T120 offline;
alter rollback segment T121 offline;
alter rollback segment T122 offline;
alter rollback segment T123 offline;
alter rollback segment T124 offline;
alter rollback segment T125 offline;
alter rollback segment T126 offline;
alter rollback segment T127 offline;
alter rollback segment T128 offline;
alter rollback segment T129 offline;
alter rollback segment T130 offline;
alter rollback segment T131 offline;
alter rollback segment T132 offline;
alter rollback segment T133 offline;
alter rollback segment T134 offline;
alter rollback segment T135 offline;
alter rollback segment T136 offline;
alter rollback segment T137 offline;
alter rollback segment T138 offline;
alter rollback segment T139 offline;
alter rollback segment T140 offline;
alter rollback segment T141 offline;
alter rollback segment T142 offline;
alter rollback segment T143 offline;
alter rollback segment T144 offline;
alter rollback segment T145 offline;
alter rollback segment T146 offline;
alter rollback segment T147 offline;
alter rollback segment T148 offline;
alter rollback segment T149 offline;
alter rollback segment T150 offline;
alter rollback segment T151 offline;
alter rollback segment T152 offline;
alter rollback segment T153 offline;
alter rollback segment T154 offline;
alter rollback segment T155 offline;
alter rollback segment T156 offline;
alter rollback segment T157 offline;
alter rollback segment T158 offline;
alter rollback segment T159 offline;
alter rollback segment T160 offline;
```

```
alter rollback segment T161 offline;
alter rollback segment T162 offline;
alter rollback segment T163 offline;
alter rollback segment T164 offline;
alter rollback segment T165 offline;
alter rollback segment T166 offline;
alter rollback segment T167 offline;
alter rollback segment T168 offline;
alter rollback segment T169 offline;
alter rollback segment T170 offline;
alter rollback segment T171 offline;
alter rollback segment T172 offline;
alter rollback segment T173 offline;
alter rollback segment T174 offline;
alter rollback segment T175 offline;
alter rollback segment T176 offline;
alter rollback segment T177 offline;
alter rollback segment T178 offline;
alter rollback segment T179 offline;
alter rollback segment T180 offline;
alter rollback segment T181 offline;
alter rollback segment T182 offline;
alter rollback segment T183 offline;
alter rollback segment T184 offline;
alter rollback segment T185 offline;
alter rollback segment T186 offline;
alter rollback segment T187 offline;
alter rollback segment T188 offline;
alter rollback segment T189 offline;
alter rollback segment T190 offline;
alter rollback segment T191 offline;
alter rollback segment T192 offline;
alter rollback segment T193 offline;
alter rollback segment T194 offline;
alter rollback segment T195 offline;
alter rollback segment T196 offline;
alter rollback segment T197 offline;
alter rollback segment T198 offline;
alter rollback segment T199 offline;
alter rollback segment T200 offline;
alter rollback segment T201 offline;
alter rollback segment T202 offline;
alter rollback segment T203 offline;
alter rollback segment T204 offline;
alter rollback segment T205 offline;
alter rollback segment T206 offline;
alter rollback segment T207 offline;
alter rollback segment T208 offline;
alter rollback segment T209 offline;
alter rollback segment T210 offline;
alter rollback segment T211 offline;
alter rollback segment T212 offline;
alter rollback segment T213 offline;
alter rollback segment T214 offline;
alter rollback segment T215 offline;
alter rollback segment T216 offline;
alter rollback segment T217 offline;
alter rollback segment T218 offline;
alter rollback segment T219 offline;
alter rollback segment T220 offline;
alter rollback segment T221 offline;
alter rollback segment T222 offline;
alter rollback segment T223 offline;
alter rollback segment T224 offline;
alter rollback segment T225 offline;
alter rollback segment T226 offline;
alter rollback segment T227 offline;
alter rollback segment T228 offline;
alter rollback segment T229 offline;
alter rollback segment T230 offline;
alter rollback segment T231 offline;
alter rollback segment T232 offline;
alter rollback segment T233 offline;
alter rollback segment T234 offline;
alter rollback segment T235 offline;
alter rollback segment T236 offline;
alter rollback segment T237 offline;
alter rollback segment T238 offline;
alter rollback segment T239 offline;
alter rollback segment T240 offline;
alter rollback segment T241 offline;
alter rollback segment T242 offline;
alter rollback segment T243 offline;
alter rollback segment T244 offline;
alter rollback segment T245 offline;
alter rollback segment T246 offline;
alter rollback segment T247 offline;
alter rollback segment T248 offline;
alter rollback segment T249 offline;
alter rollback segment T250 offline;
alter rollback segment T251 offline;
alter rollback segment T252 offline;
alter rollback segment T253 offline;
alter rollback segment T254 offline;
alter rollback segment T255 offline;
alter rollback segment T256 offline;
alter rollback segment T257 offline;
alter rollback segment T258 offline;
alter rollback segment T259 offline;
alter rollback segment T260 offline;
alter rollback segment T261 offline;
alter rollback segment T262 offline;
alter rollback segment T263 offline;
alter rollback segment T264 offline;
alter rollback segment T265 offline;
alter rollback segment T266 offline;
alter rollback segment T267 offline;
alter rollback segment T268 offline;
alter rollback segment T269 offline;
alter rollback segment T270 offline;
alter rollback segment T271 offline;
alter rollback segment T272 offline;
alter rollback segment T273 offline;
alter rollback segment T274 offline;
alter rollback segment T275 offline;
alter rollback segment T276 offline;
alter rollback segment T277 offline;
alter rollback segment T278 offline;
alter rollback segment T279 offline;
alter rollback segment T280 offline;
alter rollback segment T281 offline;
alter rollback segment T282 offline;
alter rollback segment T283 offline;
alter rollback segment T284 offline;
alter rollback segment T285 offline;
alter rollback segment T286 offline;
alter rollback segment T287 offline;
alter rollback segment T288 offline;
alter rollback segment T289 offline;
alter rollback segment T290 offline;
alter rollback segment T291 offline;
alter rollback segment T292 offline;
alter rollback segment T293 offline;
alter rollback segment T294 offline;
alter rollback segment T295 offline;
alter rollback segment T296 offline;
alter rollback segment T297 offline;
alter rollback segment T298 offline;
alter rollback segment T299 offline;
alter rollback segment T300 offline;
alter rollback segment T301 offline;
alter rollback segment T302 offline;
alter rollback segment T303 offline;
alter rollback segment T304 offline;
alter rollback segment T305 offline;
alter rollback segment T306 offline;
alter rollback segment T307 offline;
alter rollback segment T308 offline;
alter rollback segment T309 offline;
alter rollback segment T310 offline;
alter rollback segment T311 offline;
alter rollback segment T312 offline;
alter rollback segment T313 offline;
alter rollback segment T314 offline;
alter rollback segment T315 offline;
alter rollback segment T316 offline;
alter rollback segment T317 offline;
alter rollback segment T318 offline;
```

```
alter rollback segment T319 offline;
alter rollback segment T320 offline;
alter rollback segment T321 offline;
alter rollback segment T322 offline;
alter rollback segment T323 offline;
alter rollback segment T324 offline;
alter rollback segment T325 offline;
alter rollback segment T326 offline;
alter rollback segment T327 offline;
alter rollback segment T328 offline;
alter rollback segment T329 offline;
alter rollback segment T330 offline;
alter rollback segment T331 offline;
alter rollback segment T332 offline;
alter rollback segment T333 offline;
alter rollback segment T334 offline;
alter rollback segment T335 offline;
alter rollback segment T336 offline;
alter rollback segment T337 offline;
alter rollback segment T338 offline;
alter rollback segment T339 offline;
alter rollback segment T340 offline;
alter rollback segment T341 offline;
alter rollback segment T342 offline;
alter rollback segment T343 offline;
alter rollback segment T344 offline;
alter rollback segment T345 offline;
alter rollback segment T346 offline;
alter rollback segment T347 offline;
alter rollback segment T348 offline;
alter rollback segment T349 offline;
alter rollback segment T350 offline;
alter rollback segment T351 offline;
alter rollback segment T352 offline;
alter rollback segment T353 offline;
alter rollback segment T354 offline;
alter rollback segment T355 offline;
alter rollback segment T356 offline;
alter rollback segment T357 offline;
alter rollback segment T358 offline;
alter rollback segment T359 offline;
alter rollback segment T360 offline;
alter rollback segment T361 offline;
alter rollback segment T362 offline;
alter rollback segment T363 offline;
alter rollback segment T364 offline;
alter rollback segment T365 offline;
alter rollback segment T366 offline;
alter rollback segment T367 offline;
alter rollback segment T368 offline;
alter rollback segment T369 offline;
alter rollback segment T370 offline;
alter rollback segment T371 offline;
alter rollback segment T372 offline;
alter rollback segment T373 offline;
alter rollback segment T374 offline;
alter rollback segment T375 offline;
alter rollback segment T376 offline;
alter rollback segment T377 offline;
alter rollback segment T378 offline;
alter rollback segment T379 offline;
alter rollback segment T380 offline;
alter rollback segment T381 offline;
alter rollback segment T382 offline;
alter rollback segment T383 offline;
alter rollback segment T384 offline;
alter rollback segment T385 offline;
alter rollback segment T386 offline;
alter rollback segment T387 offline;
alter rollback segment T388 offline;
alter rollback segment T389 offline;
alter rollback segment T390 offline;
alter rollback segment T391 offline;
alter rollback segment T392 offline;
alter rollback segment T393 offline;
alter rollback segment T394 offline;
alter rollback segment T395 offline;
alter rollback segment T396 offline;
alter rollback segment T397 offline;
alter rollback segment T398 offline;
alter rollback segment T399 offline;
alter rollback segment T400 offline;
alter rollback segment T401 offline;
alter rollback segment T402 offline;
alter rollback segment T403 offline;
alter rollback segment T404 offline;
alter rollback segment T405 offline;
alter rollback segment T406 offline;
alter rollback segment T407 offline;
alter rollback segment T408 offline;
alter rollback segment T409 offline;
alter rollback segment T410 offline;
alter rollback segment T411 offline;
alter rollback segment T412 offline;
alter rollback segment T413 offline;
alter rollback segment T414 offline;
alter rollback segment T415 offline;
alter rollback segment T416 offline;
alter rollback segment T417 offline;
alter rollback segment T418 offline;
alter rollback segment T419 offline;
alter rollback segment T420 offline;
alter rollback segment T421 offline;
alter rollback segment T422 offline;
alter rollback segment T423 offline;
alter rollback segment T424 offline;
alter rollback segment T425 offline;
alter rollback segment T426 offline;
alter rollback segment T427 offline;
alter rollback segment T428 offline;
alter rollback segment T429 offline;
alter rollback segment T430 offline;
alter rollback segment T431 offline;
alter rollback segment T432 offline;
alter rollback segment T433 offline;
alter rollback segment T434 offline;
alter rollback segment T435 offline;
alter rollback segment T436 offline;
alter rollback segment T437 offline;
alter rollback segment T438 offline;
alter rollback segment T439 offline;
alter rollback segment T440 offline;
alter rollback segment T441 offline;
alter rollback segment T442 offline;
alter rollback segment T443 offline;
alter rollback segment T444 offline;
alter rollback segment T445 offline;
alter rollback segment T446 offline;
alter rollback segment T447 offline;
alter rollback segment T448 offline;
alter rollback segment T449 offline;
alter rollback segment T450 offline;
alter rollback segment T451 offline;
alter rollback segment T452 offline;
alter rollback segment T453 offline;
alter rollback segment T454 offline;
alter rollback segment T455 offline;
alter rollback segment T456 offline;
alter rollback segment T457 offline;
alter rollback segment T458 offline;
alter rollback segment T459 offline;
alter rollback segment T460 offline;
alter rollback segment T461 offline;
alter rollback segment T462 offline;
alter rollback segment T463 offline;
alter rollback segment T464 offline;
alter rollback segment T465 offline;
alter rollback segment T466 offline;
alter rollback segment T467 offline;
alter rollback segment T468 offline;
alter rollback segment T469 offline;
alter rollback segment T470 offline;
alter rollback segment T471 offline;
alter rollback segment T472 offline;
alter rollback segment T473 offline;
alter rollback segment T474 offline;
alter rollback segment T475 offline;
alter rollback segment T476 offline;
```

```
alter rollback segment T477 offline;
alter rollback segment T478 offline;
alter rollback segment T479 offline;
alter rollback segment T480 offline;
alter rollback segment T481 offline;
alter rollback segment T482 offline;
alter rollback segment T483 offline;
alter rollback segment T484 offline;
alter rollback segment T485 offline;
alter rollback segment T486 offline;
alter rollback segment T487 offline;
alter rollback segment T488 offline;
alter rollback segment T489 offline;
alter rollback segment T490 offline;
alter rollback segment T491 offline;
alter rollback segment T492 offline;
alter rollback segment T493 offline;
alter rollback segment T494 offline;
alter rollback segment T495 offline;
alter rollback segment T496 offline;
alter rollback segment T497 offline;
alter rollback segment T498 offline;
alter rollback segment T499 offline;
alter rollback segment T500 offline;
alter rollback segment T501 offline;
alter rollback segment T502 offline;
alter rollback segment T503 offline;
alter rollback segment T504 offline;
alter rollback segment T505 offline;
alter rollback segment T506 offline;
alter rollback segment T507 offline;
alter rollback segment T508 offline;
alter rollback segment T509 offline;
alter rollback segment T510 offline;
alter rollback segment T511 offline;
alter rollback segment T512 offline;
alter rollback segment T513 offline;
alter rollback segment T514 offline;
alter rollback segment T515 offline;
alter rollback segment T516 offline;
alter rollback segment T517 offline;
alter rollback segment T518 offline;
alter rollback segment T519 offline;
alter rollback segment T520 offline;
alter rollback segment T521 offline;
alter rollback segment T522 offline;
alter rollback segment T523 offline;
alter rollback segment T524 offline;
alter rollback segment T525 offline;
alter rollback segment T526 offline;
alter rollback segment T527 offline;
alter rollback segment T528 offline;
alter rollback segment T529 offline;
alter rollback segment T530 offline;
alter rollback segment T531 offline;
alter rollback segment T532 offline;
alter rollback segment T533 offline;
alter rollback segment T534 offline;
alter rollback segment T535 offline;
alter rollback segment T536 offline;
alter rollback segment T537 offline;
alter rollback segment T538 offline;
alter rollback segment T539 offline;
alter rollback segment T540 offline;
alter rollback segment T541 offline;
alter rollback segment T542 offline;
alter rollback segment T543 offline;
alter rollback segment T544 offline;
alter rollback segment T545 offline;
alter rollback segment T546 offline;
alter rollback segment T547 offline;
alter rollback segment T548 offline;
alter rollback segment T549 offline;
alter rollback segment T550 offline;
alter rollback segment T551 offline;
alter rollback segment T552 offline;
alter rollback segment T553 offline;
alter rollback segment T554 offline;
alter rollback segment T555 offline;
alter rollback segment T556 offline;
alter rollback segment T557 offline;
alter rollback segment T558 offline;
alter rollback segment T559 offline;
alter rollback segment T560 offline;
alter rollback segment T561 offline;
alter rollback segment T562 offline;
alter rollback segment T563 offline;
alter rollback segment T564 offline;
alter rollback segment T565 offline;
alter rollback segment T566 offline;
alter rollback segment T567 offline;
alter rollback segment T568 offline;
alter rollback segment T569 offline;
alter rollback segment T570 offline;
alter rollback segment T571 offline;
alter rollback segment T572 offline;
alter rollback segment T573 offline;
alter rollback segment T574 offline;
alter rollback segment T575 offline;
alter rollback segment T576 offline;
alter rollback segment T577 offline;
alter rollback segment T578 offline;
alter rollback segment T579 offline;
alter rollback segment T580 offline;
alter rollback segment T581 offline;
alter rollback segment T582 offline;
alter rollback segment T583 offline;
alter rollback segment T584 offline;
alter rollback segment T585 offline;
alter rollback segment T586 offline;
alter rollback segment T587 offline;
alter rollback segment T588 offline;
alter rollback segment T589 offline;
alter rollback segment T590 offline;
alter rollback segment T591 offline;
alter rollback segment T592 offline;
alter rollback segment T593 offline;
alter rollback segment T594 offline;
alter rollback segment T595 offline;
alter rollback segment T596 offline;
alter rollback segment T597 offline;
alter rollback segment T598 offline;
alter rollback segment T599 offline;
alter rollback segment T600 offline;
alter rollback segment T601 offline;
alter rollback segment T602 offline;
alter rollback segment T603 offline;
alter rollback segment T604 offline;
alter rollback segment T605 offline;
alter rollback segment T606 offline;
alter rollback segment T607 offline;
alter rollback segment T608 offline;
alter rollback segment T609 offline;
alter rollback segment T610 offline;
alter rollback segment T611 offline;
alter rollback segment T612 offline;
alter rollback segment T613 offline;
alter rollback segment T614 offline;
alter rollback segment T615 offline;
alter rollback segment T616 offline;
alter rollback segment T617 offline;
alter rollback segment T618 offline;
alter rollback segment T619 offline;
alter rollback segment T620 offline;
alter rollback segment T621 offline;
alter rollback segment T622 offline;
alter rollback segment T623 offline;
alter rollback segment T624 offline;
alter rollback segment T625 offline;
alter rollback segment T626 offline;
alter rollback segment T627 offline;
alter rollback segment T628 offline;
alter rollback segment T629 offline;
alter rollback segment T630 offline;
alter rollback segment T631 offline;
alter rollback segment T632 offline;
alter rollback segment T633 offline;
alter rollback segment T634 offline;
```

```
alter rollback segment T635 offline;
alter rollback segment T636 offline;
alter rollback segment T637 offline;
alter rollback segment T638 offline;
alter rollback segment T639 offline;
alter rollback segment T640 offline;
alter rollback segment T641 offline;
alter rollback segment T642 offline;
alter rollback segment T643 offline;
alter rollback segment T644 offline;
alter rollback segment T645 offline;
alter rollback segment T646 offline;
alter rollback segment T647 offline;
alter rollback segment T648 offline;
alter rollback segment T649 offline;
alter rollback segment T650 offline;
alter rollback segment T651 offline;
alter rollback segment T652 offline;
alter rollback segment T653 offline;
alter rollback segment T654 offline;
alter rollback segment T655 offline;
alter rollback segment T656 offline;
alter rollback segment T657 offline;
alter rollback segment T658 offline;
alter rollback segment T659 offline;
alter rollback segment T660 offline;
alter rollback segment T661 offline;
alter rollback segment T662 offline;
alter rollback segment T663 offline;
alter rollback segment T664 offline;
alter rollback segment T665 offline;
alter rollback segment T666 offline;
alter rollback segment T667 offline;
alter rollback segment T668 offline;
alter rollback segment T669 offline;
alter rollback segment T670 offline;
alter rollback segment T671 offline;
alter rollback segment T672 offline;
alter rollback segment T673 offline;
alter rollback segment T674 offline;
alter rollback segment T675 offline;
alter rollback segment T676 offline;
alter rollback segment T677 offline;
alter rollback segment T678 offline;
alter rollback segment T679 offline;
alter rollback segment T680 offline;
alter rollback segment T681 offline;
alter rollback segment T682 offline;
alter rollback segment T683 offline;
alter rollback segment T684 offline;
alter rollback segment T685 offline;
alter rollback segment T686 offline;
alter rollback segment T687 offline;
alter rollback segment T688 offline;
alter rollback segment T689 offline;
alter rollback segment T690 offline;
alter rollback segment T691 offline;
alter rollback segment T692 offline;
alter rollback segment T693 offline;
alter rollback segment T694 offline;
alter rollback segment T695 offline;
alter rollback segment T696 offline;
alter rollback segment T697 offline;
alter rollback segment T698 offline;
alter rollback segment T699 offline;
alter rollback segment T700 offline;
alter rollback segment T701 offline;
alter rollback segment T702 offline;
alter rollback segment T703 offline;
alter rollback segment T704 offline;
alter rollback segment T705 offline;
alter rollback segment T706 offline;
alter rollback segment T707 offline;
alter rollback segment T708 offline;
alter rollback segment T709 offline;
alter rollback segment T710 offline;
alter rollback segment T711 offline;
alter rollback segment T712 offline;
alter rollback segment T713 offline;
alter rollback segment T714 offline;
alter rollback segment T715 offline;
alter rollback segment T716 offline;
alter rollback segment T717 offline;
alter rollback segment T718 offline;
alter rollback segment T719 offline;
alter rollback segment T720 offline;
alter rollback segment T721 offline;
alter rollback segment T722 offline;
alter rollback segment T723 offline;
alter rollback segment T724 offline;
alter rollback segment T725 offline;
alter rollback segment T726 offline;
alter rollback segment T727 offline;
alter rollback segment T728 offline;
alter rollback segment T729 offline;
alter rollback segment T730 offline;
alter rollback segment T731 offline;
alter rollback segment T732 offline;
alter rollback segment T733 offline;
alter rollback segment T734 offline;
alter rollback segment T735 offline;
alter rollback segment T736 offline;
alter rollback segment T737 offline;
alter rollback segment T738 offline;
alter rollback segment T739 offline;
alter rollback segment T740 offline;
alter rollback segment T741 offline;
alter rollback segment T742 offline;
alter rollback segment T743 offline;
alter rollback segment T744 offline;
alter rollback segment T745 offline;
alter rollback segment T746 offline;
alter rollback segment T747 offline;
alter rollback segment T748 offline;
alter rollback segment T749 offline;
alter rollback segment T750 offline;
alter rollback segment T751 offline;
alter rollback segment T752 offline;
alter rollback segment T753 offline;
alter rollback segment T754 offline;
alter rollback segment T755 offline;
alter rollback segment T756 offline;
alter rollback segment T757 offline;
alter rollback segment T758 offline;
alter rollback segment T759 offline;
alter rollback segment T760 offline;
alter rollback segment T761 offline;
alter rollback segment T762 offline;
alter rollback segment T763 offline;
alter rollback segment T764 offline;
alter rollback segment T765 offline;
alter rollback segment T766 offline;
alter rollback segment T767 offline;
alter rollback segment T768 offline;
alter rollback segment T769 offline;
alter rollback segment T770 offline;
alter rollback segment T771 offline;
alter rollback segment T772 offline;
alter rollback segment T773 offline;
alter rollback segment T774 offline;
alter rollback segment T775 offline;
alter rollback segment T776 offline;
alter rollback segment T777 offline;
alter rollback segment T778 offline;
alter rollback segment T779 offline;
alter rollback segment T780 offline;
alter rollback segment T781 offline;
alter rollback segment T782 offline;
alter rollback segment T783 offline;
alter rollback segment T784 offline;
alter rollback segment T785 offline;
alter rollback segment T786 offline;
alter rollback segment T787 offline;
alter rollback segment T788 offline;
alter rollback segment T789 offline;
alter rollback segment T790 offline;
alter rollback segment T791 offline;
alter rollback segment T792 offline;
```

```
alter rollback segment T793 offline;
alter rollback segment T794 offline;
alter rollback segment T795 offline;
alter rollback segment T796 offline;
alter rollback segment T797 offline;
alter rollback segment T798 offline;
alter rollback segment T799 offline;
alter rollback segment T800 offline;

drop rollback segment T1;
drop rollback segment T2;
drop rollback segment T3;
drop rollback segment T4;
drop rollback segment T5;
drop rollback segment T6;
drop rollback segment T7;
drop rollback segment T8;
drop rollback segment T9;
drop rollback segment T10;
drop rollback segment T11;
drop rollback segment T12;
drop rollback segment T13;
drop rollback segment T14;
drop rollback segment T15;
drop rollback segment T16;
drop rollback segment T17;
drop rollback segment T18;
drop rollback segment T19;
drop rollback segment T20;
drop rollback segment T21;
drop rollback segment T22;
drop rollback segment T23;
drop rollback segment T24;
drop rollback segment T25;
drop rollback segment T26;
drop rollback segment T27;
drop rollback segment T28;
drop rollback segment T29;
drop rollback segment T30;
drop rollback segment T31;
drop rollback segment T32;
drop rollback segment T33;
drop rollback segment T34;
drop rollback segment T35;
drop rollback segment T36;
drop rollback segment T37;
drop rollback segment T38;
drop rollback segment T39;
drop rollback segment T40;
drop rollback segment T41;
drop rollback segment T42;
drop rollback segment T43;
drop rollback segment T44;
drop rollback segment T45;
drop rollback segment T46;
drop rollback segment T47;
drop rollback segment T48;
drop rollback segment T49;
drop rollback segment T50;
drop rollback segment T51;
drop rollback segment T52;
drop rollback segment T53;
drop rollback segment T54;
drop rollback segment T55;
drop rollback segment T56;
drop rollback segment T57;
drop rollback segment T58;
drop rollback segment T59;
drop rollback segment T60;
drop rollback segment T61;
drop rollback segment T62;
drop rollback segment T63;
drop rollback segment T64;
drop rollback segment T65;
drop rollback segment T66;
drop rollback segment T67;
drop rollback segment T68;
drop rollback segment T69;
drop rollback segment T70;
drop rollback segment T71;
drop rollback segment T72;
drop rollback segment T73;
drop rollback segment T74;
drop rollback segment T75;
drop rollback segment T76;
drop rollback segment T77;
drop rollback segment T78;
drop rollback segment T79;
drop rollback segment T80;
drop rollback segment T81;
drop rollback segment T82;
drop rollback segment T83;
drop rollback segment T84;
drop rollback segment T85;
drop rollback segment T86;
drop rollback segment T87;
drop rollback segment T88;
drop rollback segment T89;
drop rollback segment T90;
drop rollback segment T91;
drop rollback segment T92;
drop rollback segment T93;
drop rollback segment T94;
drop rollback segment T95;
drop rollback segment T96;
drop rollback segment T97;
drop rollback segment T98;
drop rollback segment T99;
drop rollback segment T100;
drop rollback segment T101;
drop rollback segment T102;
drop rollback segment T103;
drop rollback segment T104;
drop rollback segment T105;
drop rollback segment T106;
drop rollback segment T107;
drop rollback segment T108;
drop rollback segment T109;
drop rollback segment T110;
drop rollback segment T111;
drop rollback segment T112;
drop rollback segment T113;
drop rollback segment T114;
drop rollback segment T115;
drop rollback segment T116;
drop rollback segment T117;
drop rollback segment T118;
drop rollback segment T119;
drop rollback segment T120;
drop rollback segment T121;
drop rollback segment T122;
drop rollback segment T123;
drop rollback segment T124;
drop rollback segment T125;
drop rollback segment T126;
drop rollback segment T127;
drop rollback segment T128;
drop rollback segment T129;
drop rollback segment T130;
drop rollback segment T131;
drop rollback segment T132;
drop rollback segment T133;
drop rollback segment T134;
drop rollback segment T135;
drop rollback segment T136;
drop rollback segment T137;
drop rollback segment T138;
drop rollback segment T139;
drop rollback segment T140;
drop rollback segment T141;
drop rollback segment T142;
drop rollback segment T143;
drop rollback segment T144;
drop rollback segment T145;
drop rollback segment T146;
drop rollback segment T147;
drop rollback segment T148;
drop rollback segment T149;
```

```
drop rollback segment T150;
drop rollback segment T151;
drop rollback segment T152;
drop rollback segment T153;
drop rollback segment T154;
drop rollback segment T155;
drop rollback segment T156;
drop rollback segment T157;
drop rollback segment T158;
drop rollback segment T159;
drop rollback segment T160;
drop rollback segment T161;
drop rollback segment T162;
drop rollback segment T163;
drop rollback segment T164;
drop rollback segment T165;
drop rollback segment T166;
drop rollback segment T167;
drop rollback segment T168;
drop rollback segment T169;
drop rollback segment T170;
drop rollback segment T171;
drop rollback segment T172;
drop rollback segment T173;
drop rollback segment T174;
drop rollback segment T175;
drop rollback segment T176;
drop rollback segment T177;
drop rollback segment T178;
drop rollback segment T179;
drop rollback segment T180;
drop rollback segment T181;
drop rollback segment T182;
drop rollback segment T183;
drop rollback segment T184;
drop rollback segment T185;
drop rollback segment T186;
drop rollback segment T187;
drop rollback segment T188;
drop rollback segment T189;
drop rollback segment T190;
drop rollback segment T191;
drop rollback segment T192;
drop rollback segment T193;
drop rollback segment T194;
drop rollback segment T195;
drop rollback segment T196;
drop rollback segment T197;
drop rollback segment T198;
drop rollback segment T199;
drop rollback segment T200;
drop rollback segment T201;
drop rollback segment T202;
drop rollback segment T203;
drop rollback segment T204;
drop rollback segment T205;
drop rollback segment T206;
drop rollback segment T207;
drop rollback segment T208;
drop rollback segment T209;
drop rollback segment T210;
drop rollback segment T211;
drop rollback segment T212;
drop rollback segment T213;
drop rollback segment T214;
drop rollback segment T215;
drop rollback segment T216;
drop rollback segment T217;
drop rollback segment T218;
drop rollback segment T219;
drop rollback segment T220;
drop rollback segment T221;
drop rollback segment T222;
drop rollback segment T223;
drop rollback segment T224;
drop rollback segment T225;
drop rollback segment T226;
drop rollback segment T227;
drop rollback segment T228;
drop rollback segment T229;
drop rollback segment T230;
drop rollback segment T231;
drop rollback segment T232;
drop rollback segment T233;
drop rollback segment T234;
drop rollback segment T235;
drop rollback segment T236;
drop rollback segment T237;
drop rollback segment T238;
drop rollback segment T239;
drop rollback segment T240;
drop rollback segment T241;
drop rollback segment T242;
drop rollback segment T243;
drop rollback segment T244;
drop rollback segment T245;
drop rollback segment T246;
drop rollback segment T247;
drop rollback segment T248;
drop rollback segment T249;
drop rollback segment T250;
drop rollback segment T251;
drop rollback segment T252;
drop rollback segment T253;
drop rollback segment T254;
drop rollback segment T255;
drop rollback segment T256;
drop rollback segment T257;
drop rollback segment T258;
drop rollback segment T259;
drop rollback segment T260;
drop rollback segment T261;
drop rollback segment T262;
drop rollback segment T263;
drop rollback segment T264;
drop rollback segment T265;
drop rollback segment T266;
drop rollback segment T267;
drop rollback segment T268;
drop rollback segment T269;
drop rollback segment T270;
drop rollback segment T271;
drop rollback segment T272;
drop rollback segment T273;
drop rollback segment T274;
drop rollback segment T275;
drop rollback segment T276;
drop rollback segment T277;
drop rollback segment T278;
drop rollback segment T279;
drop rollback segment T280;
drop rollback segment T281;
drop rollback segment T282;
drop rollback segment T283;
drop rollback segment T284;
drop rollback segment T285;
drop rollback segment T286;
drop rollback segment T287;
drop rollback segment T288;
drop rollback segment T289;
drop rollback segment T290;
drop rollback segment T291;
drop rollback segment T292;
drop rollback segment T293;
drop rollback segment T294;
drop rollback segment T295;
drop rollback segment T296;
drop rollback segment T297;
drop rollback segment T298;
drop rollback segment T299;
drop rollback segment T300;
drop rollback segment T301;
drop rollback segment T302;
drop rollback segment T303;
drop rollback segment T304;
drop rollback segment T305;
drop rollback segment T306;
drop rollback segment T307;
```

```
drop rollback segment T308;
drop rollback segment T309;
drop rollback segment T310;
drop rollback segment T311;
drop rollback segment T312;
drop rollback segment T313;
drop rollback segment T314;
drop rollback segment T315;
drop rollback segment T316;
drop rollback segment T317;
drop rollback segment T318;
drop rollback segment T319;
drop rollback segment T320;
drop rollback segment T321;
drop rollback segment T322;
drop rollback segment T323;
drop rollback segment T324;
drop rollback segment T325;
drop rollback segment T326;
drop rollback segment T327;
drop rollback segment T328;
drop rollback segment T329;
drop rollback segment T330;
drop rollback segment T331;
drop rollback segment T332;
drop rollback segment T333;
drop rollback segment T334;
drop rollback segment T335;
drop rollback segment T336;
drop rollback segment T337;
drop rollback segment T338;
drop rollback segment T339;
drop rollback segment T340;
drop rollback segment T341;
drop rollback segment T342;
drop rollback segment T343;
drop rollback segment T344;
drop rollback segment T345;
drop rollback segment T346;
drop rollback segment T347;
drop rollback segment T348;
drop rollback segment T349;
drop rollback segment T350;
drop rollback segment T351;
drop rollback segment T352;
drop rollback segment T353;
drop rollback segment T354;
drop rollback segment T355;
drop rollback segment T356;
drop rollback segment T357;
drop rollback segment T358;
drop rollback segment T359;
drop rollback segment T360;
drop rollback segment T361;
drop rollback segment T362;
drop rollback segment T363;
drop rollback segment T364;
drop rollback segment T365;
drop rollback segment T366;
drop rollback segment T367;
drop rollback segment T368;
drop rollback segment T369;
drop rollback segment T370;
drop rollback segment T371;
drop rollback segment T372;
drop rollback segment T373;
drop rollback segment T374;
drop rollback segment T375;
drop rollback segment T376;
drop rollback segment T377;
drop rollback segment T378;
drop rollback segment T379;
drop rollback segment T380;
drop rollback segment T381;
drop rollback segment T382;
drop rollback segment T383;
drop rollback segment T384;
drop rollback segment T385;
drop rollback segment T386;
drop rollback segment T387;
drop rollback segment T388;
drop rollback segment T389;
drop rollback segment T390;
drop rollback segment T391;
drop rollback segment T392;
drop rollback segment T393;
drop rollback segment T394;
drop rollback segment T395;
drop rollback segment T396;
drop rollback segment T397;
drop rollback segment T398;
drop rollback segment T399;
drop rollback segment T400;
drop rollback segment T401;
drop rollback segment T402;
drop rollback segment T403;
drop rollback segment T404;
drop rollback segment T405;
drop rollback segment T406;
drop rollback segment T407;
drop rollback segment T408;
drop rollback segment T409;
drop rollback segment T410;
drop rollback segment T411;
drop rollback segment T412;
drop rollback segment T413;
drop rollback segment T414;
drop rollback segment T415;
drop rollback segment T416;
drop rollback segment T417;
drop rollback segment T418;
drop rollback segment T419;
drop rollback segment T420;
drop rollback segment T421;
drop rollback segment T422;
drop rollback segment T423;
drop rollback segment T424;
drop rollback segment T425;
drop rollback segment T426;
drop rollback segment T427;
drop rollback segment T428;
drop rollback segment T429;
drop rollback segment T430;
drop rollback segment T431;
drop rollback segment T432;
drop rollback segment T433;
drop rollback segment T434;
drop rollback segment T435;
drop rollback segment T436;
drop rollback segment T437;
drop rollback segment T438;
drop rollback segment T439;
drop rollback segment T440;
drop rollback segment T441;
drop rollback segment T442;
drop rollback segment T443;
drop rollback segment T444;
drop rollback segment T445;
drop rollback segment T446;
drop rollback segment T447;
drop rollback segment T448;
drop rollback segment T449;
drop rollback segment T450;
drop rollback segment T451;
drop rollback segment T452;
drop rollback segment T453;
drop rollback segment T454;
drop rollback segment T455;
drop rollback segment T456;
drop rollback segment T457;
drop rollback segment T458;
drop rollback segment T459;
drop rollback segment T460;
drop rollback segment T461;
drop rollback segment T462;
drop rollback segment T463;
drop rollback segment T464;
drop rollback segment T465;
```

```
drop rollback segment T466;
drop rollback segment T467;
drop rollback segment T468;
drop rollback segment T469;
drop rollback segment T470;
drop rollback segment T471;
drop rollback segment T472;
drop rollback segment T473;
drop rollback segment T474;
drop rollback segment T475;
drop rollback segment T476;
drop rollback segment T477;
drop rollback segment T478;
drop rollback segment T479;
drop rollback segment T480;
drop rollback segment T481;
drop rollback segment T482;
drop rollback segment T483;
drop rollback segment T484;
drop rollback segment T485;
drop rollback segment T486;
drop rollback segment T487;
drop rollback segment T488;
drop rollback segment T489;
drop rollback segment T490;
drop rollback segment T491;
drop rollback segment T492;
drop rollback segment T493;
drop rollback segment T494;
drop rollback segment T495;
drop rollback segment T496;
drop rollback segment T497;
drop rollback segment T498;
drop rollback segment T499;
drop rollback segment T500;
drop rollback segment T501;
drop rollback segment T502;
drop rollback segment T503;
drop rollback segment T504;
drop rollback segment T505;
drop rollback segment T506;
drop rollback segment T507;
drop rollback segment T508;
drop rollback segment T509;
drop rollback segment T510;
drop rollback segment T511;
drop rollback segment T512;
drop rollback segment T513;
drop rollback segment T514;
drop rollback segment T515;
drop rollback segment T516;
drop rollback segment T517;
drop rollback segment T518;
drop rollback segment T519;
drop rollback segment T520;
drop rollback segment T521;
drop rollback segment T522;
drop rollback segment T523;
drop rollback segment T524;
drop rollback segment T525;
drop rollback segment T526;
drop rollback segment T527;
drop rollback segment T528;
drop rollback segment T529;
drop rollback segment T530;
drop rollback segment T531;
drop rollback segment T532;
drop rollback segment T533;
drop rollback segment T534;
drop rollback segment T535;
drop rollback segment T536;
drop rollback segment T537;
drop rollback segment T538;
drop rollback segment T539;
drop rollback segment T540;
drop rollback segment T541;
drop rollback segment T542;
drop rollback segment T543;
drop rollback segment T544;
drop rollback segment T545;
drop rollback segment T546;
drop rollback segment T547;
drop rollback segment T548;
drop rollback segment T549;
drop rollback segment T550;
drop rollback segment T551;
drop rollback segment T552;
drop rollback segment T553;
drop rollback segment T554;
drop rollback segment T555;
drop rollback segment T556;
drop rollback segment T557;
drop rollback segment T558;
drop rollback segment T559;
drop rollback segment T560;
drop rollback segment T561;
drop rollback segment T562;
drop rollback segment T563;
drop rollback segment T564;
drop rollback segment T565;
drop rollback segment T566;
drop rollback segment T567;
drop rollback segment T568;
drop rollback segment T569;
drop rollback segment T570;
drop rollback segment T571;
drop rollback segment T572;
drop rollback segment T573;
drop rollback segment T574;
drop rollback segment T575;
drop rollback segment T576;
drop rollback segment T577;
drop rollback segment T578;
drop rollback segment T579;
drop rollback segment T580;
drop rollback segment T581;
drop rollback segment T582;
drop rollback segment T583;
drop rollback segment T584;
drop rollback segment T585;
drop rollback segment T586;
drop rollback segment T587;
drop rollback segment T588;
drop rollback segment T589;
drop rollback segment T590;
drop rollback segment T591;
drop rollback segment T592;
drop rollback segment T593;
drop rollback segment T594;
drop rollback segment T595;
drop rollback segment T596;
drop rollback segment T597;
drop rollback segment T598;
drop rollback segment T599;
drop rollback segment T600;
drop rollback segment T601;
drop rollback segment T602;
drop rollback segment T603;
drop rollback segment T604;
drop rollback segment T605;
drop rollback segment T606;
drop rollback segment T607;
drop rollback segment T608;
drop rollback segment T609;
drop rollback segment T610;
drop rollback segment T611;
drop rollback segment T612;
drop rollback segment T613;
drop rollback segment T614;
drop rollback segment T615;
drop rollback segment T616;
drop rollback segment T617;
drop rollback segment T618;
drop rollback segment T619;
drop rollback segment T620;
drop rollback segment T621;
drop rollback segment T622;
drop rollback segment T623;
```

```
drop rollback segment T624;
drop rollback segment T625;
drop rollback segment T626;
drop rollback segment T627;
drop rollback segment T628;
drop rollback segment T629;
drop rollback segment T630;
drop rollback segment T631;
drop rollback segment T632;
drop rollback segment T633;
drop rollback segment T634;
drop rollback segment T635;
drop rollback segment T636;
drop rollback segment T637;
drop rollback segment T638;
drop rollback segment T639;
drop rollback segment T640;
drop rollback segment T641;
drop rollback segment T642;
drop rollback segment T643;
drop rollback segment T644;
drop rollback segment T645;
drop rollback segment T646;
drop rollback segment T647;
drop rollback segment T648;
drop rollback segment T649;
drop rollback segment T650;
drop rollback segment T651;
drop rollback segment T652;
drop rollback segment T653;
drop rollback segment T654;
drop rollback segment T655;
drop rollback segment T656;
drop rollback segment T657;
drop rollback segment T658;
drop rollback segment T659;
drop rollback segment T660;
drop rollback segment T661;
drop rollback segment T662;
drop rollback segment T663;
drop rollback segment T664;
drop rollback segment T665;
drop rollback segment T666;
drop rollback segment T667;
drop rollback segment T668;
drop rollback segment T669;
drop rollback segment T670;
drop rollback segment T671;
drop rollback segment T672;
drop rollback segment T673;
drop rollback segment T674;
drop rollback segment T675;
drop rollback segment T676;
drop rollback segment T677;
drop rollback segment T678;
drop rollback segment T679;
drop rollback segment T680;
drop rollback segment T681;
drop rollback segment T682;
drop rollback segment T683;
drop rollback segment T684;
drop rollback segment T685;
drop rollback segment T686;
drop rollback segment T687;
drop rollback segment T688;
drop rollback segment T689;
drop rollback segment T690;
drop rollback segment T691;
drop rollback segment T692;
drop rollback segment T693;
drop rollback segment T694;
drop rollback segment T695;
drop rollback segment T696;
drop rollback segment T697;
drop rollback segment T698;
drop rollback segment T699;
drop rollback segment T700;
drop rollback segment T701;
drop rollback segment T702;
drop rollback segment T703;
drop rollback segment T704;
drop rollback segment T705;
drop rollback segment T706;
drop rollback segment T707;
drop rollback segment T708;
drop rollback segment T709;
drop rollback segment T710;
drop rollback segment T711;
drop rollback segment T712;
drop rollback segment T713;
drop rollback segment T714;
drop rollback segment T715;
drop rollback segment T716;
drop rollback segment T717;
drop rollback segment T718;
drop rollback segment T719;
drop rollback segment T720;
drop rollback segment T721;
drop rollback segment T722;
drop rollback segment T723;
drop rollback segment T724;
drop rollback segment T725;
drop rollback segment T726;
drop rollback segment T727;
drop rollback segment T728;
drop rollback segment T729;
drop rollback segment T730;
drop rollback segment T731;
drop rollback segment T732;
drop rollback segment T733;
drop rollback segment T734;
drop rollback segment T735;
drop rollback segment T736;
drop rollback segment T737;
drop rollback segment T738;
drop rollback segment T739;
drop rollback segment T740;
drop rollback segment T741;
drop rollback segment T742;
drop rollback segment T743;
drop rollback segment T744;
drop rollback segment T745;
drop rollback segment T746;
drop rollback segment T747;
drop rollback segment T748;
drop rollback segment T749;
drop rollback segment T750;
drop rollback segment T751;
drop rollback segment T752;
drop rollback segment T753;
drop rollback segment T754;
drop rollback segment T755;
drop rollback segment T756;
drop rollback segment T757;
drop rollback segment T758;
drop rollback segment T759;
drop rollback segment T760;
drop rollback segment T761;
drop rollback segment T762;
drop rollback segment T763;
drop rollback segment T764;
drop rollback segment T765;
drop rollback segment T766;
drop rollback segment T767;
drop rollback segment T768;
drop rollback segment T769;
drop rollback segment T770;
drop rollback segment T771;
drop rollback segment T772;
drop rollback segment T773;
drop rollback segment T774;
drop rollback segment T775;
drop rollback segment T776;
drop rollback segment T777;
drop rollback segment T778;
drop rollback segment T779;
drop rollback segment T780;
drop rollback segment T781;
```

```
drop rollback segment T782;
drop rollback segment T783;
drop rollback segment T784;
drop rollback segment T785;
drop rollback segment T786;
drop rollback segment T787;
drop rollback segment T788;
drop rollback segment T789;
drop rollback segment T790;
drop rollback segment T791;
drop rollback segment T792;
drop rollback segment T793;
drop rollback segment T794;
drop rollback segment T795;
drop rollback segment T796;
drop rollback segment T797;
drop rollback segment T798;
drop rollback segment T799;
drop rollback segment T800;

                    CREATE ROLLBACK SEGMENT t1 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t2 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t3 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t4 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t5 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t6 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t7 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t8 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t9 TABLESPACE roll storage (initial 200K
next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t10 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t11 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t12 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t13 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t14 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t15 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t16 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t17 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t18 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t19 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t20 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t21 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t22 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t23 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t24 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t25 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t26 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t27 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t28 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t29 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t30 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t31 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t32 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t33 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t34 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t35 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t36 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t37 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t38 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t39 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t40 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t41 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t42 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t43 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t44 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t45 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t46 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t47 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t48 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t49 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t50 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t51 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t52 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t53 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t54 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t55 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t56 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t57 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t58 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t59 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t60 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t61 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t62 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t63 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t64 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t65 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t66 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t67 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                    CREATE ROLLBACK SEGMENT t68 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t69 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t70 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t71 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t72 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t73 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t74 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t75 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t76 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t77 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t78 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t79 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t80 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t81 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t82 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t83 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t84 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t85 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t86 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t87 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t88 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t89 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t90 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t91 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t92 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t93 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t94 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t95 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t96 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t97 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t98 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t99 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t100 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t101 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t102 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t103 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t104 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t105 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t106 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t107 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t108 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t109 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t110 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t111 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t112 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t113 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t114 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t115 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t116 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t117 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t118 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t119 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t120 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t121 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t122 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t123 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t124 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t125 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t126 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t127 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t128 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t129 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t130 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t131 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t132 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t133 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t134 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t135 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t136 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t137 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t138 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t139 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t140 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t141 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t142 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t143 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t144 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t145 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t146 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t147 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t148 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t149 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t150 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t151 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t152 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t153 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t154 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t155 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t156 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t157 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t158 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t159 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t160 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t161 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t162 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t163 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t164 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t165 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t166 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t167 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t168 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t169 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t170 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t171 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t172 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t173 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t174 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t175 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t176 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t177 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t178 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t179 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t180 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t181 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t182 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t183 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t184 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t185 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t186 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t187 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t188 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t189 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t190 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t191 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t192 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t193 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t194 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t195 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t196 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t197 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t198 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t199 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t200 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t201 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t202 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t203 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t204 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t205 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t206 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t207 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t208 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t209 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t210 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t211 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t212 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t213 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t214 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t215 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t216 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t217 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t218 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t219 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t220 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t221 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t222 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t223 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t224 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t225 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t226 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t227 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t228 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t229 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t230 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t231 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t232 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t233 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t234 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t235 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t236 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t237 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t238 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t239 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t240 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t241 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t242 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t243 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t244 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t245 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t246 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t247 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t248 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t249 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t250 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t251 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t252 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t253 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t254 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t255 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t256 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t257 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t258 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t259 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t260 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t261 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t262 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t263 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t264 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t265 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t266 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t267 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t268 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t269 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t270 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t271 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t272 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t273 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t274 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t275 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t276 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t277 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t278 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t279 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t280 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t281 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t282 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t283 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t284 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t285 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t286 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t287 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t288 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t289 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t290 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t291 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t292 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t293 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t294 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t295 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t296 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t297 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t298 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t299 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t300 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t301 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t302 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t303 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t304 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t305 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t306 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t307 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t308 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t309 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t310 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t311 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t312 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t313 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t314 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t315 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t316 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t317 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t318 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t319 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t320 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t321 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t322 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t323 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t324 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t325 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t326 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t327 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t328 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t329 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t330 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t331 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t332 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t333 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t334 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t335 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t336 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t337 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t338 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t339 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t340 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t341 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t342 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t343 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t344 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t345 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t346 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t347 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t348 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t349 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t350 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t351 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t352 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t353 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t354 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t355 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t356 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t357 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t358 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t359 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t360 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t361 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t362 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t363 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t364 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t365 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t366 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t367 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t368 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t369 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t370 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t371 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t372 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t373 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t374 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t375 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t376 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t377 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t378 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t379 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t380 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t381 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t382 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t383 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t384 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t385 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t386 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t387 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t388 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t389 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t390 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t391 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t392 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t393 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t394 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t395 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t396 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t397 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t398 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t399 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t400 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t401 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t402 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t403 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t404 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t405 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t406 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t407 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t408 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t409 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t410 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t411 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t412 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t413 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t414 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t415 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t416 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t417 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t418 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t419 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t420 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t421 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t422 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t423 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t424 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t425 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t426 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t427 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t428 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t429 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t430 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t431 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t432 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t433 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t434 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t435 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t436 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t437 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t438 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t439 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t440 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t441 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t442 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t443 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t444 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t445 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t446 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t447 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t448 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t449 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t450 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t451 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t452 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t453 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t454 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t455 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t456 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t457 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t458 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t459 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t460 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t461 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t462 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t463 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t464 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t465 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t466 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t467 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t468 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t469 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t470 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t471 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t472 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t473 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t474 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t475 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t476 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t477 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t478 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t479 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t480 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t481 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t482 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t483 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t484 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t485 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t486 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t487 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t488 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t489 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t490 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t491 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t492 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t493 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t494 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t495 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t496 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t497 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t498 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t499 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t500 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t501 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t502 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t503 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t504 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t505 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t506 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t507 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t508 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t509 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t510 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t511 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t512 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t513 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t514 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t515 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t516 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t517 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t518 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t519 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t520 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t521 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t522 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t523 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t524 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t525 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t526 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t527 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t528 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t529 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t530 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t531 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t532 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t533 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t534 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t535 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t536 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t537 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t538 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t539 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t540 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t541 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t542 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t543 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t544 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t545 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t546 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t547 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t548 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t549 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t550 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t551 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t552 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t553 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t554 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t555 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t556 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t557 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t558 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t559 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t560 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t561 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t562 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t563 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t564 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t565 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t566 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t567 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t568 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t569 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t570 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t571 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t572 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t573 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t574 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t575 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t576 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t577 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t578 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t579 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t580 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t581 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t582 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t583 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t584 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t585 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t586 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t587 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t588 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t589 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t590 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t591 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t592 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t593 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t594 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t595 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t596 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t597 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t598 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t599 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t600 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t601 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t602 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t603 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t604 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t605 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t606 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t607 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t608 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t609 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t610 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t611 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t612 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t613 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t614 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t615 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t616 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t617 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t618 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t619 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t620 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t621 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t622 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t623 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t624 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t625 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t626 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t627 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t628 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t629 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t630 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t631 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t632 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t633 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t634 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t635 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t636 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t637 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t638 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t639 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t640 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t641 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t642 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t643 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t644 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t645 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t646 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t647 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t648 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t649 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t650 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t651 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t652 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t653 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t654 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t655 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t656 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t657 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t658 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t659 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t660 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t661 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t662 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t663 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t664 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t665 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t666 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t667 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t668 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t669 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t670 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t671 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t672 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t673 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t674 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t675 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t676 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t677 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t678 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t679 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t680 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t681 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t682 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t683 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t684 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t685 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t686 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t687 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t688 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t689 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t690 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t691 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t692 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t693 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t694 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t695 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t696 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t697 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t698 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t699 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t700 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t701 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t702 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t703 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t704 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t705 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t706 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t707 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t708 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t709 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t710 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t711 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t712 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t713 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t714 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t715 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t716 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t717 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t718 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t719 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t720 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t721 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t722 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t723 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t724 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t725 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t726 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t727 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t728 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t729 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t730 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t731 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t732 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t733 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t734 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t735 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t736 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t737 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t738 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t739 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t740 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t741 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t742 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t743 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t744 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t745 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t746 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t747 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t748 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t749 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t750 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t751 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t752 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t753 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t754 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t755 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t756 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t757 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t758 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t759 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t760 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t761 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t762 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t763 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t764 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t765 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t766 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t767 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t768 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t769 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t770 TABLESPACE roll storage (initial
200K next 200K minextents 2);
```

```
                CREATE ROLLBACK SEGMENT t771 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t772 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t773 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t774 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t775 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t776 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t777 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t778 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t779 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t780 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t781 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t782 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t783 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t784 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t785 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t786 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t787 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t788 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t789 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t790 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t791 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t792 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t793 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t794 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t795 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t796 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t797 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t798 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t799 TABLESPACE roll storage (initial
200K next 200K minextents 2);
                CREATE ROLLBACK SEGMENT t800 TABLESPACE roll storage (initial
200K next 200K minextents 2);
alter rollback segment T1 online;
alter rollback segment T2 online;
alter rollback segment T3 online;
alter rollback segment T4 online;
alter rollback segment T5 online;
alter rollback segment T6 online;
alter rollback segment T7 online;
alter rollback segment T8 online;
alter rollback segment T9 online;
alter rollback segment T10 online;
alter rollback segment T11 online;
alter rollback segment T12 online;
alter rollback segment T13 online;
alter rollback segment T14 online;
alter rollback segment T15 online;
alter rollback segment T16 online;
alter rollback segment T17 online;
alter rollback segment T18 online;
alter rollback segment T19 online;
alter rollback segment T20 online;
alter rollback segment T21 online;
alter rollback segment T22 online;
alter rollback segment T23 online;
alter rollback segment T24 online;
alter rollback segment T25 online;
alter rollback segment T26 online;
alter rollback segment T27 online;
alter rollback segment T28 online;
alter rollback segment T29 online;
alter rollback segment T30 online;
alter rollback segment T31 online;
alter rollback segment T32 online;
alter rollback segment T33 online;
alter rollback segment T34 online;
alter rollback segment T35 online;
alter rollback segment T36 online;
alter rollback segment T37 online;
alter rollback segment T38 online;
alter rollback segment T39 online;
alter rollback segment T40 online;
alter rollback segment T41 online;
alter rollback segment T42 online;
alter rollback segment T43 online;
alter rollback segment T44 online;
alter rollback segment T45 online;
alter rollback segment T46 online;
alter rollback segment T47 online;
alter rollback segment T48 online;
alter rollback segment T49 online;
alter rollback segment T50 online;
alter rollback segment T51 online;
alter rollback segment T52 online;
alter rollback segment T53 online;
alter rollback segment T54 online;
alter rollback segment T55 online;
alter rollback segment T56 online;
alter rollback segment T57 online;
alter rollback segment T58 online;
alter rollback segment T59 online;
alter rollback segment T60 online;
alter rollback segment T61 online;
alter rollback segment T62 online;
alter rollback segment T63 online;
alter rollback segment T64 online;
alter rollback segment T65 online;
alter rollback segment T66 online;
alter rollback segment T67 online;
alter rollback segment T68 online;
alter rollback segment T69 online;
alter rollback segment T70 online;
alter rollback segment T71 online;
alter rollback segment T72 online;
alter rollback segment T73 online;
alter rollback segment T74 online;
alter rollback segment T75 online;
alter rollback segment T76 online;
alter rollback segment T77 online;
alter rollback segment T78 online;
alter rollback segment T79 online;
alter rollback segment T80 online;
alter rollback segment T81 online;
alter rollback segment T82 online;
alter rollback segment T83 online;
alter rollback segment T84 online;
alter rollback segment T85 online;
alter rollback segment T86 online;
alter rollback segment T87 online;
alter rollback segment T88 online;
alter rollback segment T89 online;
alter rollback segment T90 online;
alter rollback segment T91 online;
alter rollback segment T92 online;
alter rollback segment T93 online;
alter rollback segment T94 online;
alter rollback segment T95 online;
alter rollback segment T96 online;
alter rollback segment T97 online;
alter rollback segment T98 online;
```

```
alter rollback segment T99 online;
alter rollback segment T100 online;
alter rollback segment T101 online;
alter rollback segment T102 online;
alter rollback segment T103 online;
alter rollback segment T104 online;
alter rollback segment T105 online;
alter rollback segment T106 online;
alter rollback segment T107 online;
alter rollback segment T108 online;
alter rollback segment T109 online;
alter rollback segment T110 online;
alter rollback segment T111 online;
alter rollback segment T112 online;
alter rollback segment T113 online;
alter rollback segment T114 online;
alter rollback segment T115 online;
alter rollback segment T116 online;
alter rollback segment T117 online;
alter rollback segment T118 online;
alter rollback segment T119 online;
alter rollback segment T120 online;
alter rollback segment T121 online;
alter rollback segment T122 online;
alter rollback segment T123 online;
alter rollback segment T124 online;
alter rollback segment T125 online;
alter rollback segment T126 online;
alter rollback segment T127 online;
alter rollback segment T128 online;
alter rollback segment T129 online;
alter rollback segment T130 online;
alter rollback segment T131 online;
alter rollback segment T132 online;
alter rollback segment T133 online;
alter rollback segment T134 online;
alter rollback segment T135 online;
alter rollback segment T136 online;
alter rollback segment T137 online;
alter rollback segment T138 online;
alter rollback segment T139 online;
alter rollback segment T140 online;
alter rollback segment T141 online;
alter rollback segment T142 online;
alter rollback segment T143 online;
alter rollback segment T144 online;
alter rollback segment T145 online;
alter rollback segment T146 online;
alter rollback segment T147 online;
alter rollback segment T148 online;
alter rollback segment T149 online;
alter rollback segment T150 online;
alter rollback segment T151 online;
alter rollback segment T152 online;
alter rollback segment T153 online;
alter rollback segment T154 online;
alter rollback segment T155 online;
alter rollback segment T156 online;
alter rollback segment T157 online;
alter rollback segment T158 online;
alter rollback segment T159 online;
alter rollback segment T160 online;
alter rollback segment T161 online;
alter rollback segment T162 online;
alter rollback segment T163 online;
alter rollback segment T164 online;
alter rollback segment T165 online;
alter rollback segment T166 online;
alter rollback segment T167 online;
alter rollback segment T168 online;
alter rollback segment T169 online;
alter rollback segment T170 online;
alter rollback segment T171 online;
alter rollback segment T172 online;
alter rollback segment T173 online;
alter rollback segment T174 online;
alter rollback segment T175 online;
alter rollback segment T176 online;
alter rollback segment T177 online;
alter rollback segment T178 online;
alter rollback segment T179 online;
alter rollback segment T180 online;
alter rollback segment T181 online;
alter rollback segment T182 online;
alter rollback segment T183 online;
alter rollback segment T184 online;
alter rollback segment T185 online;
alter rollback segment T186 online;
alter rollback segment T187 online;
alter rollback segment T188 online;
alter rollback segment T189 online;
alter rollback segment T190 online;
alter rollback segment T191 online;
alter rollback segment T192 online;
alter rollback segment T193 online;
alter rollback segment T194 online;
alter rollback segment T195 online;
alter rollback segment T196 online;
alter rollback segment T197 online;
alter rollback segment T198 online;
alter rollback segment T199 online;
alter rollback segment T200 online;
alter rollback segment T201 online;
alter rollback segment T202 online;
alter rollback segment T203 online;
alter rollback segment T204 online;
alter rollback segment T205 online;
alter rollback segment T206 online;
alter rollback segment T207 online;
alter rollback segment T208 online;
alter rollback segment T209 online;
alter rollback segment T210 online;
alter rollback segment T211 online;
alter rollback segment T212 online;
alter rollback segment T213 online;
alter rollback segment T214 online;
alter rollback segment T215 online;
alter rollback segment T216 online;
alter rollback segment T217 online;
alter rollback segment T218 online;
alter rollback segment T219 online;
alter rollback segment T220 online;
alter rollback segment T221 online;
alter rollback segment T222 online;
alter rollback segment T223 online;
alter rollback segment T224 online;
alter rollback segment T225 online;
alter rollback segment T226 online;
alter rollback segment T227 online;
alter rollback segment T228 online;
alter rollback segment T229 online;
alter rollback segment T230 online;
alter rollback segment T231 online;
alter rollback segment T232 online;
alter rollback segment T233 online;
alter rollback segment T234 online;
alter rollback segment T235 online;
alter rollback segment T236 online;
alter rollback segment T237 online;
alter rollback segment T238 online;
alter rollback segment T239 online;
alter rollback segment T240 online;
alter rollback segment T241 online;
alter rollback segment T242 online;
alter rollback segment T243 online;
alter rollback segment T244 online;
alter rollback segment T245 online;
alter rollback segment T246 online;
alter rollback segment T247 online;
alter rollback segment T248 online;
alter rollback segment T249 online;
alter rollback segment T250 online;
alter rollback segment T251 online;
alter rollback segment T252 online;
alter rollback segment T253 online;
alter rollback segment T254 online;
alter rollback segment T255 online;
alter rollback segment T256 online;
```

```
alter rollback segment T257 online;
alter rollback segment T258 online;
alter rollback segment T259 online;
alter rollback segment T260 online;
alter rollback segment T261 online;
alter rollback segment T262 online;
alter rollback segment T263 online;
alter rollback segment T264 online;
alter rollback segment T265 online;
alter rollback segment T266 online;
alter rollback segment T267 online;
alter rollback segment T268 online;
alter rollback segment T269 online;
alter rollback segment T270 online;
alter rollback segment T271 online;
alter rollback segment T272 online;
alter rollback segment T273 online;
alter rollback segment T274 online;
alter rollback segment T275 online;
alter rollback segment T276 online;
alter rollback segment T277 online;
alter rollback segment T278 online;
alter rollback segment T279 online;
alter rollback segment T280 online;
alter rollback segment T281 online;
alter rollback segment T282 online;
alter rollback segment T283 online;
alter rollback segment T284 online;
alter rollback segment T285 online;
alter rollback segment T286 online;
alter rollback segment T287 online;
alter rollback segment T288 online;
alter rollback segment T289 online;
alter rollback segment T290 online;
alter rollback segment T291 online;
alter rollback segment T292 online;
alter rollback segment T293 online;
alter rollback segment T294 online;
alter rollback segment T295 online;
alter rollback segment T296 online;
alter rollback segment T297 online;
alter rollback segment T298 online;
alter rollback segment T299 online;
alter rollback segment T300 online;
alter rollback segment T301 online;
alter rollback segment T302 online;
alter rollback segment T303 online;
alter rollback segment T304 online;
alter rollback segment T305 online;
alter rollback segment T306 online;
alter rollback segment T307 online;
alter rollback segment T308 online;
alter rollback segment T309 online;
alter rollback segment T310 online;
alter rollback segment T311 online;
alter rollback segment T312 online;
alter rollback segment T313 online;
alter rollback segment T314 online;
alter rollback segment T315 online;
alter rollback segment T316 online;
alter rollback segment T317 online;
alter rollback segment T318 online;
alter rollback segment T319 online;
alter rollback segment T320 online;
alter rollback segment T321 online;
alter rollback segment T322 online;
alter rollback segment T323 online;
alter rollback segment T324 online;
alter rollback segment T325 online;
alter rollback segment T326 online;
alter rollback segment T327 online;
alter rollback segment T328 online;
alter rollback segment T329 online;
alter rollback segment T330 online;
alter rollback segment T331 online;
alter rollback segment T332 online;
alter rollback segment T333 online;
alter rollback segment T334 online;
alter rollback segment T335 online;
alter rollback segment T336 online;
alter rollback segment T337 online;
alter rollback segment T338 online;
alter rollback segment T339 online;
alter rollback segment T340 online;
alter rollback segment T341 online;
alter rollback segment T342 online;
alter rollback segment T343 online;
alter rollback segment T344 online;
alter rollback segment T345 online;
alter rollback segment T346 online;
alter rollback segment T347 online;
alter rollback segment T348 online;
alter rollback segment T349 online;
alter rollback segment T350 online;
alter rollback segment T351 online;
alter rollback segment T352 online;
alter rollback segment T353 online;
alter rollback segment T354 online;
alter rollback segment T355 online;
alter rollback segment T356 online;
alter rollback segment T357 online;
alter rollback segment T358 online;
alter rollback segment T359 online;
alter rollback segment T360 online;
alter rollback segment T361 online;
alter rollback segment T362 online;
alter rollback segment T363 online;
alter rollback segment T364 online;
alter rollback segment T365 online;
alter rollback segment T366 online;
alter rollback segment T367 online;
alter rollback segment T368 online;
alter rollback segment T369 online;
alter rollback segment T370 online;
alter rollback segment T371 online;
alter rollback segment T372 online;
alter rollback segment T373 online;
alter rollback segment T374 online;
alter rollback segment T375 online;
alter rollback segment T376 online;
alter rollback segment T377 online;
alter rollback segment T378 online;
alter rollback segment T379 online;
alter rollback segment T380 online;
alter rollback segment T381 online;
alter rollback segment T382 online;
alter rollback segment T383 online;
alter rollback segment T384 online;
alter rollback segment T385 online;
alter rollback segment T386 online;
alter rollback segment T387 online;
alter rollback segment T388 online;
alter rollback segment T389 online;
alter rollback segment T390 online;
alter rollback segment T391 online;
alter rollback segment T392 online;
alter rollback segment T393 online;
alter rollback segment T394 online;
alter rollback segment T395 online;
alter rollback segment T396 online;
alter rollback segment T397 online;
alter rollback segment T398 online;
alter rollback segment T399 online;
alter rollback segment T400 online;
alter rollback segment T401 online;
alter rollback segment T402 online;
alter rollback segment T403 online;
alter rollback segment T404 online;
alter rollback segment T405 online;
alter rollback segment T406 online;
alter rollback segment T407 online;
alter rollback segment T408 online;
alter rollback segment T409 online;
alter rollback segment T410 online;
alter rollback segment T411 online;
alter rollback segment T412 online;
alter rollback segment T413 online;
alter rollback segment T414 online;
```

```
alter rollback segment T415 online;
alter rollback segment T416 online;
alter rollback segment T417 online;
alter rollback segment T418 online;
alter rollback segment T419 online;
alter rollback segment T420 online;
alter rollback segment T421 online;
alter rollback segment T422 online;
alter rollback segment T423 online;
alter rollback segment T424 online;
alter rollback segment T425 online;
alter rollback segment T426 online;
alter rollback segment T427 online;
alter rollback segment T428 online;
alter rollback segment T429 online;
alter rollback segment T430 online;
alter rollback segment T431 online;
alter rollback segment T432 online;
alter rollback segment T433 online;
alter rollback segment T434 online;
alter rollback segment T435 online;
alter rollback segment T436 online;
alter rollback segment T437 online;
alter rollback segment T438 online;
alter rollback segment T439 online;
alter rollback segment T440 online;
alter rollback segment T441 online;
alter rollback segment T442 online;
alter rollback segment T443 online;
alter rollback segment T444 online;
alter rollback segment T445 online;
alter rollback segment T446 online;
alter rollback segment T447 online;
alter rollback segment T448 online;
alter rollback segment T449 online;
alter rollback segment T450 online;
alter rollback segment T451 online;
alter rollback segment T452 online;
alter rollback segment T453 online;
alter rollback segment T454 online;
alter rollback segment T455 online;
alter rollback segment T456 online;
alter rollback segment T457 online;
alter rollback segment T458 online;
alter rollback segment T459 online;
alter rollback segment T460 online;
alter rollback segment T461 online;
alter rollback segment T462 online;
alter rollback segment T463 online;
alter rollback segment T464 online;
alter rollback segment T465 online;
alter rollback segment T466 online;
alter rollback segment T467 online;
alter rollback segment T468 online;
alter rollback segment T469 online;
alter rollback segment T470 online;
alter rollback segment T471 online;
alter rollback segment T472 online;
alter rollback segment T473 online;
alter rollback segment T474 online;
alter rollback segment T475 online;
alter rollback segment T476 online;
alter rollback segment T477 online;
alter rollback segment T478 online;
alter rollback segment T479 online;
alter rollback segment T480 online;
alter rollback segment T481 online;
alter rollback segment T482 online;
alter rollback segment T483 online;
alter rollback segment T484 online;
alter rollback segment T485 online;
alter rollback segment T486 online;
alter rollback segment T487 online;
alter rollback segment T488 online;
alter rollback segment T489 online;
alter rollback segment T490 online;
alter rollback segment T491 online;
alter rollback segment T492 online;
alter rollback segment T493 online;
alter rollback segment T494 online;
alter rollback segment T495 online;
alter rollback segment T496 online;
alter rollback segment T497 online;
alter rollback segment T498 online;
alter rollback segment T499 online;
alter rollback segment T500 online;
alter rollback segment T501 online;
alter rollback segment T502 online;
alter rollback segment T503 online;
alter rollback segment T504 online;
alter rollback segment T505 online;
alter rollback segment T506 online;
alter rollback segment T507 online;
alter rollback segment T508 online;
alter rollback segment T509 online;
alter rollback segment T510 online;
alter rollback segment T511 online;
alter rollback segment T512 online;
alter rollback segment T513 online;
alter rollback segment T514 online;
alter rollback segment T515 online;
alter rollback segment T516 online;
alter rollback segment T517 online;
alter rollback segment T518 online;
alter rollback segment T519 online;
alter rollback segment T520 online;
alter rollback segment T521 online;
alter rollback segment T522 online;
alter rollback segment T523 online;
alter rollback segment T524 online;
alter rollback segment T525 online;
alter rollback segment T526 online;
alter rollback segment T527 online;
alter rollback segment T528 online;
alter rollback segment T529 online;
alter rollback segment T530 online;
alter rollback segment T531 online;
alter rollback segment T532 online;
alter rollback segment T533 online;
alter rollback segment T534 online;
alter rollback segment T535 online;
alter rollback segment T536 online;
alter rollback segment T537 online;
alter rollback segment T538 online;
alter rollback segment T539 online;
alter rollback segment T540 online;
alter rollback segment T541 online;
alter rollback segment T542 online;
alter rollback segment T543 online;
alter rollback segment T544 online;
alter rollback segment T545 online;
alter rollback segment T546 online;
alter rollback segment T547 online;
alter rollback segment T548 online;
alter rollback segment T549 online;
alter rollback segment T550 online;
alter rollback segment T551 online;
alter rollback segment T552 online;
alter rollback segment T553 online;
alter rollback segment T554 online;
alter rollback segment T555 online;
alter rollback segment T556 online;
alter rollback segment T557 online;
alter rollback segment T558 online;
alter rollback segment T559 online;
alter rollback segment T560 online;
alter rollback segment T561 online;
alter rollback segment T562 online;
alter rollback segment T563 online;
alter rollback segment T564 online;
alter rollback segment T565 online;
alter rollback segment T566 online;
alter rollback segment T567 online;
alter rollback segment T568 online;
alter rollback segment T569 online;
alter rollback segment T570 online;
alter rollback segment T571 online;
alter rollback segment T572 online;
```

```
alter rollback segment T573 online;
alter rollback segment T574 online;
alter rollback segment T575 online;
alter rollback segment T576 online;
alter rollback segment T577 online;
alter rollback segment T578 online;
alter rollback segment T579 online;
alter rollback segment T580 online;
alter rollback segment T581 online;
alter rollback segment T582 online;
alter rollback segment T583 online;
alter rollback segment T584 online;
alter rollback segment T585 online;
alter rollback segment T586 online;
alter rollback segment T587 online;
alter rollback segment T588 online;
alter rollback segment T589 online;
alter rollback segment T590 online;
alter rollback segment T591 online;
alter rollback segment T592 online;
alter rollback segment T593 online;
alter rollback segment T594 online;
alter rollback segment T595 online;
alter rollback segment T596 online;
alter rollback segment T597 online;
alter rollback segment T598 online;
alter rollback segment T599 online;
alter rollback segment T600 online;
alter rollback segment T601 online;
alter rollback segment T602 online;
alter rollback segment T603 online;
alter rollback segment T604 online;
alter rollback segment T605 online;
alter rollback segment T606 online;
alter rollback segment T607 online;
alter rollback segment T608 online;
alter rollback segment T609 online;
alter rollback segment T610 online;
alter rollback segment T611 online;
alter rollback segment T612 online;
alter rollback segment T613 online;
alter rollback segment T614 online;
alter rollback segment T615 online;
alter rollback segment T616 online;
alter rollback segment T617 online;
alter rollback segment T618 online;
alter rollback segment T619 online;
alter rollback segment T620 online;
alter rollback segment T621 online;
alter rollback segment T622 online;
alter rollback segment T623 online;
alter rollback segment T624 online;
alter rollback segment T625 online;
alter rollback segment T626 online;
alter rollback segment T627 online;
alter rollback segment T628 online;
alter rollback segment T629 online;
alter rollback segment T630 online;
alter rollback segment T631 online;
alter rollback segment T632 online;
alter rollback segment T633 online;
alter rollback segment T634 online;
alter rollback segment T635 online;
alter rollback segment T636 online;
alter rollback segment T637 online;
alter rollback segment T638 online;
alter rollback segment T639 online;
alter rollback segment T640 online;
alter rollback segment T641 online;
alter rollback segment T642 online;
alter rollback segment T643 online;
alter rollback segment T644 online;
alter rollback segment T645 online;
alter rollback segment T646 online;
alter rollback segment T647 online;
alter rollback segment T648 online;
alter rollback segment T649 online;
alter rollback segment T650 online;
alter rollback segment T651 online;
alter rollback segment T652 online;
alter rollback segment T653 online;
alter rollback segment T654 online;
alter rollback segment T655 online;
alter rollback segment T656 online;
alter rollback segment T657 online;
alter rollback segment T658 online;
alter rollback segment T659 online;
alter rollback segment T660 online;
alter rollback segment T661 online;
alter rollback segment T662 online;
alter rollback segment T663 online;
alter rollback segment T664 online;
alter rollback segment T665 online;
alter rollback segment T666 online;
alter rollback segment T667 online;
alter rollback segment T668 online;
alter rollback segment T669 online;
alter rollback segment T670 online;
alter rollback segment T671 online;
alter rollback segment T672 online;
alter rollback segment T673 online;
alter rollback segment T674 online;
alter rollback segment T675 online;
alter rollback segment T676 online;
alter rollback segment T677 online;
alter rollback segment T678 online;
alter rollback segment T679 online;
alter rollback segment T680 online;
alter rollback segment T681 online;
alter rollback segment T682 online;
alter rollback segment T683 online;
alter rollback segment T684 online;
alter rollback segment T685 online;
alter rollback segment T686 online;
alter rollback segment T687 online;
alter rollback segment T688 online;
alter rollback segment T689 online;
alter rollback segment T690 online;
alter rollback segment T691 online;
alter rollback segment T692 online;
alter rollback segment T693 online;
alter rollback segment T694 online;
alter rollback segment T695 online;
alter rollback segment T696 online;
alter rollback segment T697 online;
alter rollback segment T698 online;
alter rollback segment T699 online;
alter rollback segment T700 online;
alter rollback segment T701 online;
alter rollback segment T702 online;
alter rollback segment T703 online;
alter rollback segment T704 online;
alter rollback segment T705 online;
alter rollback segment T706 online;
alter rollback segment T707 online;
alter rollback segment T708 online;
alter rollback segment T709 online;
alter rollback segment T710 online;
alter rollback segment T711 online;
alter rollback segment T712 online;
alter rollback segment T713 online;
alter rollback segment T714 online;
alter rollback segment T715 online;
alter rollback segment T716 online;
alter rollback segment T717 online;
alter rollback segment T718 online;
alter rollback segment T719 online;
alter rollback segment T720 online;
alter rollback segment T721 online;
alter rollback segment T722 online;
alter rollback segment T723 online;
alter rollback segment T724 online;
alter rollback segment T725 online;
alter rollback segment T726 online;
alter rollback segment T727 online;
alter rollback segment T728 online;
alter rollback segment T729 online;
alter rollback segment T730 online;
```

```
alter rollback segment T731 online;
alter rollback segment T732 online;
alter rollback segment T733 online;
alter rollback segment T734 online;
alter rollback segment T735 online;
alter rollback segment T736 online;
alter rollback segment T737 online;
alter rollback segment T738 online;
alter rollback segment T739 online;
alter rollback segment T740 online;
alter rollback segment T741 online;
alter rollback segment T742 online;
alter rollback segment T743 online;
alter rollback segment T744 online;
alter rollback segment T745 online;
alter rollback segment T746 online;
alter rollback segment T747 online;
alter rollback segment T748 online;
alter rollback segment T749 online;
alter rollback segment T750 online;
alter rollback segment T751 online;
alter rollback segment T752 online;
alter rollback segment T753 online;
alter rollback segment T754 online;
alter rollback segment T755 online;
alter rollback segment T756 online;
alter rollback segment T757 online;
alter rollback segment T758 online;
alter rollback segment T759 online;
alter rollback segment T760 online;
alter rollback segment T761 online;
alter rollback segment T762 online;
alter rollback segment T763 online;
alter rollback segment T764 online;
alter rollback segment T765 online;
alter rollback segment T766 online;
alter rollback segment T767 online;
alter rollback segment T768 online;
alter rollback segment T769 online;
alter rollback segment T770 online;
alter rollback segment T771 online;
alter rollback segment T772 online;
alter rollback segment T773 online;
alter rollback segment T774 online;
alter rollback segment T775 online;
alter rollback segment T776 online;
alter rollback segment T777 online;
alter rollback segment T778 online;
alter rollback segment T779 online;
alter rollback segment T780 online;
alter rollback segment T781 online;
alter rollback segment T782 online;
alter rollback segment T783 online;
alter rollback segment T784 online;
alter rollback segment T785 online;
alter rollback segment T786 online;
alter rollback segment T787 online;
alter rollback segment T788 online;
alter rollback segment T789 online;
alter rollback segment T790 online;
alter rollback segment T791 online;
alter rollback segment T792 online;
alter rollback segment T793 online;
alter rollback segment T794 online;
alter rollback segment T795 online;
alter rollback segment T796 online;
alter rollback segment T797 online;
alter rollback segment T798 online;
alter rollback segment T799 online;
alter rollback segment T800 online;

alter rollback segment S1 offline;
alter rollback segment S2 offline;
alter rollback segment S3 offline;
alter rollback segment S4 offline;
alter rollback segment S5 offline;
alter rollback segment S6 offline;
alter rollback segment S7 offline;
alter rollback segment S8 offline;
alter rollback segment S9 offline;
alter rollback segment S10 offline;
alter rollback segment S11 offline;
alter rollback segment S12 offline;
alter rollback segment S13 offline;
alter rollback segment S14 offline;
alter rollback segment S15 offline;
alter rollback segment S16 offline;
alter rollback segment S17 offline;
alter rollback segment S18 offline;
alter rollback segment S19 offline;
alter rollback segment S20 offline;
alter rollback segment S21 offline;
alter rollback segment S22 offline;
alter rollback segment S23 offline;
alter rollback segment S24 offline;
alter rollback segment S25 offline;
alter rollback segment S26 offline;
alter rollback segment S27 offline;
alter rollback segment S28 offline;
alter rollback segment S29 offline;
alter rollback segment S30 offline;
exit;
!
```

build/scripts/stock_keep.sh

```
sqlplus tpcc/tpcc <<!
alter table stock enable table lock;
alter table item  enable table lock;
alter cluster icluster storage ( buffer_pool keep );
alter cluster scluster  storage ( buffer_pool keep );
alter table stock disable table lock;
alter table item disable table lock;
exit;
!
```

build/scripts/tpcc.rol.h

```
#!/bin/ksh
#
#
#================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA     |
#                  OPEN SYSTEMS PERFORMANCE GROUP                |
#                       All Rights Reserved                      |
#================================================================+
# FILENAME
#      tpccrol.sh
# DESCRIPTION
#      Script file for creating the roll;back segments.
#================================================================
#

svrmgrl <<!
   connect internal;
   host date;
   set timing on;


                    CREATE ROLLBACK SEGMENT t1 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t2 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t3 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t4 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t5 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t6 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t7 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t8 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t9 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t10 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t11 TABLESPACE roll;
                    CREATE ROLLBACK SEGMENT t12 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t13 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t14 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t15 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t16 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t17 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t18 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t19 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t20 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t21 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t22 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t23 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t24 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t25 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t26 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t27 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t28 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t29 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t30 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t31 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t32 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t33 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t34 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t35 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t36 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t37 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t38 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t39 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t40 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t41 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t42 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t43 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t44 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t45 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t46 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t47 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t48 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t49 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t50 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t51 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t52 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t53 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t54 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t55 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t56 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t57 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t58 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t59 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t60 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t61 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t62 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t63 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t64 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t65 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t66 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t67 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t68 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t69 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t70 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t71 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t72 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t73 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t74 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t75 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t76 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t77 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t78 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t79 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t80 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t81 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t82 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t83 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t84 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t85 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t86 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t87 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t88 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t89 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t90 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t91 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t92 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t93 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t94 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t95 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t96 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t97 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t98 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t99 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t100 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t101 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t102 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t103 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t104 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t105 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t106 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t107 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t108 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t109 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t110 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t111 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t112 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t113 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t114 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t115 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t116 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t117 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t118 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t119 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t120 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t121 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t122 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t123 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t124 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t125 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t126 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t127 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t128 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t129 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t130 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t131 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t132 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t133 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t134 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t135 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t136 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t137 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t138 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t139 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t140 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t141 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t142 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t143 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t144 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t145 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t146 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t147 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t148 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t149 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t150 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t151 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t152 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t153 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t154 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t155 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t156 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t157 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t158 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t159 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t160 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t161 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t162 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t163 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t164 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t165 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t166 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t167 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t168 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t169 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t170 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t171 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t172 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t173 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t174 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t175 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t176 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t177 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t178 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t179 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t180 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t181 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t182 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t183 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t184 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t185 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t186 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t187 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t188 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t189 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t190 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t191 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t192 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t193 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t194 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t195 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t196 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t197 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t198 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t199 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t200 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t201 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t202 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t203 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t204 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t205 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t206 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t207 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t208 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t209 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t210 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t211 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t212 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t213 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t214 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t215 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t216 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t217 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t218 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t219 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t220 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t221 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t222 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t223 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t224 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t225 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t226 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t227 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t228 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t229 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t230 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t231 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t232 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t233 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t234 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t235 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t236 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t237 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t238 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t239 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t240 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t241 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t242 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t243 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t244 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t245 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t246 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t247 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t248 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t249 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t250 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t251 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t252 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t253 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t254 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t255 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t256 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t257 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t258 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t259 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t260 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t261 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t262 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t263 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t264 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t265 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t266 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t267 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t268 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t269 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t270 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t271 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t272 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t273 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t274 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t275 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t276 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t277 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t278 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t279 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t280 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t281 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t282 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t283 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t284 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t285 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t286 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t287 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t288 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t289 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t290 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t291 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t292 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t293 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t294 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t295 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t296 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t297 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t298 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t299 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t300 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t301 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t302 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t303 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t304 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t305 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t306 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t307 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t308 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t309 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t310 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t311 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t312 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t313 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t314 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t315 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t316 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t317 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t318 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t319 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t320 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t321 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t322 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t323 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t324 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t325 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t326 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t327 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t328 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t329 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t330 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t331 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t332 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t333 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t334 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t335 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t336 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t337 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t338 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t339 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t340 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t341 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t342 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t343 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t344 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t345 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t346 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t347 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t348 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t349 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t350 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t351 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t352 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t353 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t354 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t355 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t356 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t357 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t358 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t359 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t360 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t361 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t362 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t363 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t364 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t365 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t366 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t367 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t368 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t369 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t370 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t371 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t372 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t373 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t374 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t375 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t376 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t377 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t378 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t379 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t380 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t381 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t382 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t383 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t384 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t385 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t386 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t387 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t388 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t389 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t390 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t391 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t392 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t393 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t394 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t395 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t396 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t397 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t398 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t399 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t400 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t401 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t402 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t403 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t404 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t405 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t406 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t407 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t408 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t409 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t410 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t411 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t412 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t413 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t414 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t415 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t416 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t417 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t418 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t419 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t420 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t421 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t422 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t423 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t424 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t425 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t426 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t427 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t428 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t429 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t430 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t431 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t432 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t433 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t434 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t435 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t436 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t437 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t438 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t439 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t440 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t441 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t442 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t443 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t444 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t445 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t446 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t447 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t448 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t449 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t450 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t451 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t452 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t453 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t454 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t455 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t456 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t457 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t458 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t459 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t460 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t461 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t462 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t463 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t464 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t465 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t466 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t467 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t468 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t469 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t470 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t471 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t472 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t473 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t474 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t475 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t476 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t477 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t478 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t479 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t480 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t481 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t482 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t483 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t484 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t485 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t486 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t487 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t488 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t489 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t490 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t491 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t492 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t493 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t494 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t495 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t496 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t497 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t498 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t499 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t500 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t501 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t502 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t503 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t504 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t505 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t506 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t507 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t508 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t509 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t510 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t511 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t512 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t513 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t514 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t515 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t516 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t517 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t518 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t519 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t520 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t521 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t522 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t523 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t524 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t525 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t526 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t527 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t528 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t529 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t530 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t531 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t532 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t533 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t534 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t535 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t536 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t537 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t538 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t539 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t540 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t541 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t542 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t543 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t544 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t545 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t546 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t547 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t548 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t549 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t550 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t551 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t552 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t553 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t554 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t555 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t556 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t557 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t558 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t559 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t560 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t561 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t562 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t563 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t564 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t565 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t566 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t567 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t568 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t569 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t570 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t571 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t572 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t573 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t574 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t575 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t576 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t577 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t578 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t579 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t580 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t581 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t582 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t583 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t584 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t585 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t586 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t587 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t588 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t589 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t590 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t591 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t592 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t593 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t594 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t595 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t596 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t597 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t598 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t599 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t600 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t601 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t602 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t603 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t604 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t605 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t606 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t607 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t608 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t609 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t610 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t611 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t612 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t613 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t614 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t615 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t616 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t617 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t618 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t619 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t620 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t621 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t622 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t623 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t624 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t625 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t626 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t627 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t628 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t629 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t630 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t631 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t632 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t633 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t634 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t635 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t636 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t637 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t638 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t639 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t640 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t641 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t642 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t643 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t644 TABLESPACE roll;
```

```
CREATE ROLLBACK SEGMENT t645 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t646 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t647 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t648 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t649 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t650 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t651 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t652 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t653 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t654 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t655 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t656 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t657 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t658 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t659 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t660 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t661 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t662 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t663 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t664 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t665 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t666 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t667 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t668 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t669 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t670 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t671 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t672 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t673 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t674 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t675 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t676 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t677 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t678 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t679 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t680 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t681 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t682 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t683 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t684 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t685 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t686 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t687 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t688 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t689 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t690 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t691 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t692 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t693 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t694 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t695 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t696 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t697 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t698 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t699 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t700 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t701 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t702 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t703 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t704 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t705 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t706 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t707 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t708 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t709 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t710 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t711 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t712 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t713 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t714 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t715 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t716 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t717 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t718 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t719 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t720 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t721 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t722 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t723 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t724 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t725 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t726 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t727 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t728 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t729 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t730 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t731 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t732 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t733 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t734 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t735 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t736 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t737 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t738 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t739 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t740 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t741 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t742 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t743 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t744 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t745 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t746 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t747 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t748 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t749 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t750 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t751 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t752 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t753 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t754 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t755 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t756 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t757 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t758 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t759 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t760 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t761 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t762 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t763 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t764 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t765 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t766 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t767 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t768 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t769 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t770 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t771 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t772 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t773 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t774 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t775 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t776 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t777 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t778 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t779 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t780 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t781 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t782 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t783 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t784 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t785 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t786 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t787 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t788 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t789 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t790 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t791 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t792 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t793 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t794 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t795 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t796 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t797 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t798 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t799 TABLESPACE roll;
CREATE ROLLBACK SEGMENT t800 TABLESPACE roll;
!
```

build/scripts/tpcc_stored_proc.sh

```
#!/bin/ksh

# benchsetup 80301 98/6/1 16:17 vmakhija
# Copyright (c) 1998 Oracle
#
#
#=================================================================+
#        Copyright (c) 1998  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# NAME
#      benchsetup
# DESCRIPTION
#      Usage: benchsetup.sh [options]
#=================================================================
#

. build.env

sqlplus tpcc/tpcc @$TPCC_BLOCKS/views
sqlplus tpcc/tpcc @$TPCC_BLOCKS/initpay
sqlplus tpcc/tpcc @$TPCC_BLOCKS/tkvcinin
```

build/utils/dml.sh

```
#
# $Header: dml.sh 7030100.1 96/05/02 10:22:52 plai Generic<base> $ Copyr (c) 1995
Oracle
#

#=================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#     dml.sh
# DESCRIPTION
#     Disable table locks for TPC-C tables.
# USAGE
#     dml.sh
#=================================================================*/

sqlplus tpcc/tpcc <<!
   alter table warehouse  disable table lock;
   alter table district   disable table lock;
   alter table customer   disable table lock;
   alter table history    disable table lock;
   alter table item       disable table lock;
   alter table stock      disable table lock;
   alter table orders     disable table lock;
   alter table new_order  disable table lock;
   alter table order_line disable table lock;
   quit;
!
```

build/utils/undml.sh

```
#
# $Header: undml.sh 7030100.2 96/05/02 10:29:30 plai Generic<base> $ Copyr (c) 1995
Oracle
#

#=================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA      |
#                    OPEN SYSTEMS PERFORMANCE GROUP               |
#                         All Rights Reserved                     |
#=================================================================+
# FILENAME
#     undml.sh
# DESCRIPTION
#     Enable table locks for TPC-C tables.
# USAGE
#     undml.sh
#=================================================================*/

sqlplus tpcc/tpcc <<!
   alter table warehouse  enable table lock;
   alter table district   enable table lock;
   alter table customer   enable table lock;
   alter table history    enable table lock;
   alter table item       enable table lock;
   alter table stock      enable table lock;
   alter table orders     enable table lock;
   alter table new_order  enable table lock;
   alter table order_line enable table lock;
   quit;
!
```

10.1 SQL Subsidiary Scripts

build/alterroll.sql

```
alter rollback segment T1 online;
alter rollback segment T2 online;
alter rollback segment T3 online;
alter rollback segment T4 online;
alter rollback segment T5 online;
alter rollback segment T6 online;
alter rollback segment T7 online;
alter rollback segment T8 online;
alter rollback segment T9 online;
alter rollback segment T10 online;
alter rollback segment T11 online;
alter rollback segment T12 online;
alter rollback segment T13 online;
alter rollback segment T14 online;
alter rollback segment T15 online;
alter rollback segment T16 online;
alter rollback segment T17 online;
alter rollback segment T18 online;
alter rollback segment T19 online;
alter rollback segment T20 online;
alter rollback segment T21 online;
alter rollback segment T22 online;
alter rollback segment T23 online;
alter rollback segment T24 online;
alter rollback segment T25 online;
alter rollback segment T26 online;
alter rollback segment T27 online;
alter rollback segment T28 online;
alter rollback segment T29 online;
alter rollback segment T30 online;
alter rollback segment T31 online;
alter rollback segment T32 online;
alter rollback segment T33 online;
alter rollback segment T34 online;
alter rollback segment T35 online;
alter rollback segment T36 online;
alter rollback segment T37 online;
alter rollback segment T38 online;
alter rollback segment T39 online;
alter rollback segment T40 online;
alter rollback segment T41 online;
alter rollback segment T42 online;
alter rollback segment T43 online;
```

```
alter rollback segment T44 online;
alter rollback segment T45 online;
alter rollback segment T46 online;
alter rollback segment T47 online;
alter rollback segment T48 online;
alter rollback segment T49 online;
alter rollback segment T50 online;
alter rollback segment T51 online;
alter rollback segment T52 online;
alter rollback segment T53 online;
alter rollback segment T54 online;
alter rollback segment T55 online;
alter rollback segment T56 online;
alter rollback segment T57 online;
alter rollback segment T58 online;
alter rollback segment T59 online;
alter rollback segment T60 online;
alter rollback segment T61 online;
alter rollback segment T62 online;
alter rollback segment T63 online;
alter rollback segment T64 online;
alter rollback segment T65 online;
alter rollback segment T66 online;
alter rollback segment T67 online;
alter rollback segment T68 online;
alter rollback segment T69 online;
alter rollback segment T70 online;
alter rollback segment T71 online;
alter rollback segment T72 online;
alter rollback segment T73 online;
alter rollback segment T74 online;
alter rollback segment T75 online;
alter rollback segment T76 online;
alter rollback segment T77 online;
alter rollback segment T78 online;
alter rollback segment T79 online;
alter rollback segment T80 online;
alter rollback segment T81 online;
alter rollback segment T82 online;
alter rollback segment T83 online;
alter rollback segment T84 online;
alter rollback segment T85 online;
alter rollback segment T86 online;
alter rollback segment T87 online;
alter rollback segment T88 online;
alter rollback segment T89 online;
alter rollback segment T90 online;
alter rollback segment T91 online;
alter rollback segment T92 online;
alter rollback segment T93 online;
alter rollback segment T94 online;
alter rollback segment T95 online;
alter rollback segment T96 online;
alter rollback segment T97 online;
alter rollback segment T98 online;
alter rollback segment T99 online;
alter rollback segment T100 online;
alter rollback segment T101 online;
alter rollback segment T102 online;
alter rollback segment T103 online;
alter rollback segment T104 online;
alter rollback segment T105 online;
alter rollback segment T106 online;
alter rollback segment T107 online;
alter rollback segment T108 online;
alter rollback segment T109 online;
alter rollback segment T110 online;
alter rollback segment T111 online;
alter rollback segment T112 online;
alter rollback segment T113 online;
alter rollback segment T114 online;
alter rollback segment T115 online;
alter rollback segment T116 online;
alter rollback segment T117 online;
alter rollback segment T118 online;
alter rollback segment T119 online;
alter rollback segment T120 online;
alter rollback segment T121 online;
alter rollback segment T122 online;
alter rollback segment T123 online;
alter rollback segment T124 online;
alter rollback segment T125 online;
alter rollback segment T126 online;
alter rollback segment T127 online;
alter rollback segment T128 online;
alter rollback segment T129 online;
alter rollback segment T130 online;
alter rollback segment T131 online;
alter rollback segment T132 online;
alter rollback segment T133 online;
alter rollback segment T134 online;
alter rollback segment T135 online;
alter rollback segment T136 online;
alter rollback segment T137 online;
alter rollback segment T138 online;
alter rollback segment T139 online;
alter rollback segment T140 online;
alter rollback segment T141 online;
alter rollback segment T142 online;
alter rollback segment T143 online;
alter rollback segment T144 online;
alter rollback segment T145 online;
alter rollback segment T146 online;
alter rollback segment T147 online;
alter rollback segment T148 online;
alter rollback segment T149 online;
alter rollback segment T150 online;
alter rollback segment T151 online;
alter rollback segment T152 online;
alter rollback segment T153 online;
alter rollback segment T154 online;
alter rollback segment T155 online;
alter rollback segment T156 online;
alter rollback segment T157 online;
alter rollback segment T158 online;
alter rollback segment T159 online;
alter rollback segment T160 online;
alter rollback segment T161 online;
alter rollback segment T162 online;
alter rollback segment T163 online;
alter rollback segment T164 online;
alter rollback segment T165 online;
alter rollback segment T166 online;
alter rollback segment T167 online;
alter rollback segment T168 online;
alter rollback segment T169 online;
alter rollback segment T170 online;
alter rollback segment T171 online;
alter rollback segment T172 online;
alter rollback segment T173 online;
alter rollback segment T174 online;
alter rollback segment T175 online;
alter rollback segment T176 online;
alter rollback segment T177 online;
alter rollback segment T178 online;
alter rollback segment T179 online;
alter rollback segment T180 online;
alter rollback segment T181 online;
alter rollback segment T182 online;
alter rollback segment T183 online;
alter rollback segment T184 online;
alter rollback segment T185 online;
alter rollback segment T186 online;
alter rollback segment T187 online;
alter rollback segment T188 online;
alter rollback segment T189 online;
alter rollback segment T190 online;
alter rollback segment T191 online;
alter rollback segment T192 online;
alter rollback segment T193 online;
alter rollback segment T194 online;
alter rollback segment T195 online;
alter rollback segment T196 online;
alter rollback segment T197 online;
alter rollback segment T198 online;
alter rollback segment T199 online;
alter rollback segment T200 online;
alter rollback segment T201 online;
```

```
alter rollback segment T202 online;
alter rollback segment T203 online;
alter rollback segment T204 online;
alter rollback segment T205 online;
alter rollback segment T206 online;
alter rollback segment T207 online;
alter rollback segment T208 online;
alter rollback segment T209 online;
alter rollback segment T210 online;
alter rollback segment T211 online;
alter rollback segment T212 online;
alter rollback segment T213 online;
alter rollback segment T214 online;
alter rollback segment T215 online;
alter rollback segment T216 online;
alter rollback segment T217 online;
alter rollback segment T218 online;
alter rollback segment T219 online;
alter rollback segment T220 online;
alter rollback segment T221 online;
alter rollback segment T222 online;
alter rollback segment T223 online;
alter rollback segment T224 online;
alter rollback segment T225 online;
alter rollback segment T226 online;
alter rollback segment T227 online;
alter rollback segment T228 online;
alter rollback segment T229 online;
alter rollback segment T230 online;
alter rollback segment T231 online;
alter rollback segment T232 online;
alter rollback segment T233 online;
alter rollback segment T234 online;
alter rollback segment T235 online;
alter rollback segment T236 online;
alter rollback segment T237 online;
alter rollback segment T238 online;
alter rollback segment T239 online;
alter rollback segment T240 online;
alter rollback segment T241 online;
alter rollback segment T242 online;
alter rollback segment T243 online;
alter rollback segment T244 online;
alter rollback segment T245 online;
alter rollback segment T246 online;
alter rollback segment T247 online;
alter rollback segment T248 online;
alter rollback segment T249 online;
alter rollback segment T250 online;
alter rollback segment T251 online;
alter rollback segment T252 online;
alter rollback segment T253 online;
alter rollback segment T254 online;
alter rollback segment T255 online;
alter rollback segment T256 online;
alter rollback segment T257 online;
alter rollback segment T258 online;
alter rollback segment T259 online;
alter rollback segment T260 online;
alter rollback segment T261 online;
alter rollback segment T262 online;
alter rollback segment T263 online;
alter rollback segment T264 online;
alter rollback segment T265 online;
alter rollback segment T266 online;
alter rollback segment T267 online;
alter rollback segment T268 online;
alter rollback segment T269 online;
alter rollback segment T270 online;
alter rollback segment T271 online;
alter rollback segment T272 online;
alter rollback segment T273 online;
alter rollback segment T274 online;
alter rollback segment T275 online;
alter rollback segment T276 online;
alter rollback segment T277 online;
alter rollback segment T278 online;
alter rollback segment T279 online;
alter rollback segment T280 online;
alter rollback segment T281 online;
alter rollback segment T282 online;
alter rollback segment T283 online;
alter rollback segment T284 online;
alter rollback segment T285 online;
alter rollback segment T286 online;
alter rollback segment T287 online;
alter rollback segment T288 online;
alter rollback segment T289 online;
alter rollback segment T290 online;
alter rollback segment T291 online;
alter rollback segment T292 online;
alter rollback segment T293 online;
alter rollback segment T294 online;
alter rollback segment T295 online;
alter rollback segment T296 online;
alter rollback segment T297 online;
alter rollback segment T298 online;
alter rollback segment T299 online;
alter rollback segment T300 online;
alter rollback segment T301 online;
alter rollback segment T302 online;
alter rollback segment T303 online;
alter rollback segment T304 online;
alter rollback segment T305 online;
alter rollback segment T306 online;
alter rollback segment T307 online;
alter rollback segment T308 online;
alter rollback segment T309 online;
alter rollback segment T310 online;
alter rollback segment T311 online;
alter rollback segment T312 online;
alter rollback segment T313 online;
alter rollback segment T314 online;
alter rollback segment T315 online;
alter rollback segment T316 online;
alter rollback segment T317 online;
alter rollback segment T318 online;
alter rollback segment T319 online;
alter rollback segment T320 online;
alter rollback segment T321 online;
alter rollback segment T322 online;
alter rollback segment T323 online;
alter rollback segment T324 online;
alter rollback segment T325 online;
alter rollback segment T326 online;
alter rollback segment T327 online;
alter rollback segment T328 online;
alter rollback segment T329 online;
alter rollback segment T330 online;
alter rollback segment T331 online;
alter rollback segment T332 online;
alter rollback segment T333 online;
alter rollback segment T334 online;
alter rollback segment T335 online;
alter rollback segment T336 online;
alter rollback segment T337 online;
alter rollback segment T338 online;
alter rollback segment T339 online;
alter rollback segment T340 online;
alter rollback segment T341 online;
alter rollback segment T342 online;
alter rollback segment T343 online;
alter rollback segment T344 online;
alter rollback segment T345 online;
alter rollback segment T346 online;
alter rollback segment T347 online;
alter rollback segment T348 online;
alter rollback segment T349 online;
alter rollback segment T350 online;
alter rollback segment T351 online;
alter rollback segment T352 online;
alter rollback segment T353 online;
alter rollback segment T354 online;
alter rollback segment T355 online;
alter rollback segment T356 online;
alter rollback segment T357 online;
alter rollback segment T358 online;
alter rollback segment T359 online;
```

```
alter rollback segment T360 online;
alter rollback segment T361 online;
alter rollback segment T362 online;
alter rollback segment T363 online;
alter rollback segment T364 online;
alter rollback segment T365 online;
alter rollback segment T366 online;
alter rollback segment T367 online;
alter rollback segment T368 online;
alter rollback segment T369 online;
alter rollback segment T370 online;
alter rollback segment T371 online;
alter rollback segment T372 online;
alter rollback segment T373 online;
alter rollback segment T374 online;
alter rollback segment T375 online;
alter rollback segment T376 online;
alter rollback segment T377 online;
alter rollback segment T378 online;
alter rollback segment T379 online;
alter rollback segment T380 online;
alter rollback segment T381 online;
alter rollback segment T382 online;
alter rollback segment T383 online;
alter rollback segment T384 online;
alter rollback segment T385 online;
alter rollback segment T386 online;
alter rollback segment T387 online;
alter rollback segment T388 online;
alter rollback segment T389 online;
alter rollback segment T390 online;
alter rollback segment T391 online;
alter rollback segment T392 online;
alter rollback segment T393 online;
alter rollback segment T394 online;
alter rollback segment T395 online;
alter rollback segment T396 online;
alter rollback segment T397 online;
alter rollback segment T398 online;
alter rollback segment T399 online;
alter rollback segment T400 online;
alter rollback segment T401 online;
alter rollback segment T402 online;
alter rollback segment T403 online;
alter rollback segment T404 online;
alter rollback segment T405 online;
alter rollback segment T406 online;
alter rollback segment T407 online;
alter rollback segment T408 online;
alter rollback segment T409 online;
alter rollback segment T410 online;
alter rollback segment T411 online;
alter rollback segment T412 online;
alter rollback segment T413 online;
alter rollback segment T414 online;
alter rollback segment T415 online;
alter rollback segment T416 online;
alter rollback segment T417 online;
alter rollback segment T418 online;
alter rollback segment T419 online;
alter rollback segment T420 online;
alter rollback segment T421 online;
alter rollback segment T422 online;
alter rollback segment T423 online;
alter rollback segment T424 online;
alter rollback segment T425 online;
alter rollback segment T426 online;
alter rollback segment T427 online;
alter rollback segment T428 online;
alter rollback segment T429 online;
alter rollback segment T430 online;
alter rollback segment T431 online;
alter rollback segment T432 online;
alter rollback segment T433 online;
alter rollback segment T434 online;
alter rollback segment T435 online;
alter rollback segment T436 online;
alter rollback segment T437 online;
alter rollback segment T438 online;
alter rollback segment T439 online;
alter rollback segment T440 online;
alter rollback segment T441 online;
alter rollback segment T442 online;
alter rollback segment T443 online;
alter rollback segment T444 online;
alter rollback segment T445 online;
alter rollback segment T446 online;
alter rollback segment T447 online;
alter rollback segment T448 online;
alter rollback segment T449 online;
alter rollback segment T450 online;
alter rollback segment T451 online;
alter rollback segment T452 online;
alter rollback segment T453 online;
alter rollback segment T454 online;
alter rollback segment T455 online;
alter rollback segment T456 online;
alter rollback segment T457 online;
alter rollback segment T458 online;
alter rollback segment T459 online;
alter rollback segment T460 online;
alter rollback segment T461 online;
alter rollback segment T462 online;
alter rollback segment T463 online;
alter rollback segment T464 online;
alter rollback segment T465 online;
alter rollback segment T466 online;
alter rollback segment T467 online;
alter rollback segment T468 online;
alter rollback segment T469 online;
alter rollback segment T470 online;
alter rollback segment T471 online;
alter rollback segment T472 online;
alter rollback segment T473 online;
alter rollback segment T474 online;
alter rollback segment T475 online;
alter rollback segment T476 online;
alter rollback segment T477 online;
alter rollback segment T478 online;
alter rollback segment T479 online;
alter rollback segment T480 online;
alter rollback segment T481 online;
alter rollback segment T482 online;
alter rollback segment T483 online;
alter rollback segment T484 online;
alter rollback segment T485 online;
alter rollback segment T486 online;
alter rollback segment T487 online;
alter rollback segment T488 online;
alter rollback segment T489 online;
alter rollback segment T490 online;
alter rollback segment T491 online;
alter rollback segment T492 online;
alter rollback segment T493 online;
alter rollback segment T494 online;
alter rollback segment T495 online;
alter rollback segment T496 online;
alter rollback segment T497 online;
alter rollback segment T498 online;
alter rollback segment T499 online;
alter rollback segment T500 online;
alter rollback segment T501 online;
alter rollback segment T502 online;
alter rollback segment T503 online;
alter rollback segment T504 online;
alter rollback segment T505 online;
alter rollback segment T506 online;
alter rollback segment T507 online;
alter rollback segment T508 online;
alter rollback segment T509 online;
alter rollback segment T510 online;
alter rollback segment T511 online;
alter rollback segment T512 online;
alter rollback segment T513 online;
alter rollback segment T514 online;
alter rollback segment T515 online;
alter rollback segment T516 online;
alter rollback segment T517 online;
```

```
alter rollback segment T518 online;
alter rollback segment T519 online;
alter rollback segment T520 online;
alter rollback segment T521 online;
alter rollback segment T522 online;
alter rollback segment T523 online;
alter rollback segment T524 online;
alter rollback segment T525 online;
alter rollback segment T526 online;
alter rollback segment T527 online;
alter rollback segment T528 online;
alter rollback segment T529 online;
alter rollback segment T530 online;
alter rollback segment T531 online;
alter rollback segment T532 online;
alter rollback segment T533 online;
alter rollback segment T534 online;
alter rollback segment T535 online;
alter rollback segment T536 online;
alter rollback segment T537 online;
alter rollback segment T538 online;
alter rollback segment T539 online;
alter rollback segment T540 online;
alter rollback segment T541 online;
alter rollback segment T542 online;
alter rollback segment T543 online;
alter rollback segment T544 online;
alter rollback segment T545 online;
alter rollback segment T546 online;
alter rollback segment T547 online;
alter rollback segment T548 online;
alter rollback segment T549 online;
alter rollback segment T550 online;
alter rollback segment T551 online;
alter rollback segment T552 online;
alter rollback segment T553 online;
alter rollback segment T554 online;
alter rollback segment T555 online;
alter rollback segment T556 online;
alter rollback segment T557 online;
alter rollback segment T558 online;
alter rollback segment T559 online;
alter rollback segment T560 online;
alter rollback segment T561 online;
alter rollback segment T562 online;
alter rollback segment T563 online;
alter rollback segment T564 online;
alter rollback segment T565 online;
alter rollback segment T566 online;
alter rollback segment T567 online;
alter rollback segment T568 online;
alter rollback segment T569 online;
alter rollback segment T570 online;
alter rollback segment T571 online;
alter rollback segment T572 online;
alter rollback segment T573 online;
alter rollback segment T574 online;
alter rollback segment T575 online;
alter rollback segment T576 online;
alter rollback segment T577 online;
alter rollback segment T578 online;
alter rollback segment T579 online;
alter rollback segment T580 online;
alter rollback segment T581 online;
alter rollback segment T582 online;
alter rollback segment T583 online;
alter rollback segment T584 online;
alter rollback segment T585 online;
alter rollback segment T586 online;
alter rollback segment T587 online;
alter rollback segment T588 online;
alter rollback segment T589 online;
alter rollback segment T590 online;
alter rollback segment T591 online;
alter rollback segment T592 online;
alter rollback segment T593 online;
alter rollback segment T594 online;
alter rollback segment T595 online;
alter rollback segment T596 online;
alter rollback segment T597 online;
alter rollback segment T598 online;
alter rollback segment T599 online;
alter rollback segment T600 online;
alter rollback segment T601 online;
alter rollback segment T602 online;
alter rollback segment T603 online;
alter rollback segment T604 online;
alter rollback segment T605 online;
alter rollback segment T606 online;
alter rollback segment T607 online;
alter rollback segment T608 online;
alter rollback segment T609 online;
alter rollback segment T610 online;
alter rollback segment T611 online;
alter rollback segment T612 online;
alter rollback segment T613 online;
alter rollback segment T614 online;
alter rollback segment T615 online;
alter rollback segment T616 online;
alter rollback segment T617 online;
alter rollback segment T618 online;
alter rollback segment T619 online;
alter rollback segment T620 online;
alter rollback segment T621 online;
alter rollback segment T622 online;
alter rollback segment T623 online;
alter rollback segment T624 online;
alter rollback segment T625 online;
alter rollback segment T626 online;
alter rollback segment T627 online;
alter rollback segment T628 online;
alter rollback segment T629 online;
alter rollback segment T630 online;
alter rollback segment T631 online;
alter rollback segment T632 online;
alter rollback segment T633 online;
alter rollback segment T634 online;
alter rollback segment T635 online;
alter rollback segment T636 online;
alter rollback segment T637 online;
alter rollback segment T638 online;
alter rollback segment T639 online;
alter rollback segment T640 online;
alter rollback segment T641 online;
alter rollback segment T642 online;
alter rollback segment T643 online;
alter rollback segment T644 online;
alter rollback segment T645 online;
alter rollback segment T646 online;
alter rollback segment T647 online;
alter rollback segment T648 online;
alter rollback segment T649 online;
alter rollback segment T650 online;
alter rollback segment T651 online;
alter rollback segment T652 online;
alter rollback segment T653 online;
alter rollback segment T654 online;
alter rollback segment T655 online;
alter rollback segment T656 online;
alter rollback segment T657 online;
alter rollback segment T658 online;
alter rollback segment T659 online;
alter rollback segment T660 online;
alter rollback segment T661 online;
alter rollback segment T662 online;
alter rollback segment T663 online;
alter rollback segment T664 online;
alter rollback segment T665 online;
alter rollback segment T666 online;
alter rollback segment T667 online;
alter rollback segment T668 online;
alter rollback segment T669 online;
alter rollback segment T670 online;
alter rollback segment T671 online;
alter rollback segment T672 online;
alter rollback segment T673 online;
alter rollback segment T674 online;
alter rollback segment T675 online;
```

```
alter rollback segment T676 online;
alter rollback segment T677 online;
alter rollback segment T678 online;
alter rollback segment T679 online;
alter rollback segment T680 online;
alter rollback segment T681 online;
alter rollback segment T682 online;
alter rollback segment T683 online;
alter rollback segment T684 online;
alter rollback segment T685 online;
alter rollback segment T686 online;
alter rollback segment T687 online;
alter rollback segment T688 online;
alter rollback segment T689 online;
alter rollback segment T690 online;
alter rollback segment T691 online;
alter rollback segment T692 online;
alter rollback segment T693 online;
alter rollback segment T694 online;
alter rollback segment T695 online;
alter rollback segment T696 online;
alter rollback segment T697 online;
alter rollback segment T698 online;
alter rollback segment T699 online;
alter rollback segment T700 online;
alter rollback segment T701 online;
alter rollback segment T702 online;
alter rollback segment T703 online;
alter rollback segment T704 online;
alter rollback segment T705 online;
alter rollback segment T706 online;
alter rollback segment T707 online;
alter rollback segment T708 online;
alter rollback segment T709 online;
alter rollback segment T710 online;
alter rollback segment T711 online;
alter rollback segment T712 online;
alter rollback segment T713 online;
alter rollback segment T714 online;
alter rollback segment T715 online;
alter rollback segment T716 online;
alter rollback segment T717 online;
alter rollback segment T718 online;
alter rollback segment T719 online;
alter rollback segment T720 online;
alter rollback segment T721 online;
alter rollback segment T722 online;
alter rollback segment T723 online;
alter rollback segment T724 online;
alter rollback segment T725 online;
alter rollback segment T726 online;
alter rollback segment T727 online;
alter rollback segment T728 online;
alter rollback segment T729 online;
alter rollback segment T730 online;
alter rollback segment T731 online;
alter rollback segment T732 online;
alter rollback segment T733 online;
alter rollback segment T734 online;
alter rollback segment T735 online;
alter rollback segment T736 online;
alter rollback segment T737 online;
alter rollback segment T738 online;
alter rollback segment T739 online;
alter rollback segment T740 online;
alter rollback segment T741 online;
alter rollback segment T742 online;
alter rollback segment T743 online;
alter rollback segment T744 online;
alter rollback segment T745 online;
alter rollback segment T746 online;
alter rollback segment T747 online;
alter rollback segment T748 online;
alter rollback segment T749 online;
alter rollback segment T750 online;
alter rollback segment T751 online;
alter rollback segment T752 online;
alter rollback segment T753 online;
alter rollback segment T754 online;
alter rollback segment T755 online;
alter rollback segment T756 online;
alter rollback segment T757 online;
alter rollback segment T758 online;
alter rollback segment T759 online;
alter rollback segment T760 online;
alter rollback segment T761 online;
alter rollback segment T762 online;
alter rollback segment T763 online;
alter rollback segment T764 online;
alter rollback segment T765 online;
alter rollback segment T766 online;
alter rollback segment T767 online;
alter rollback segment T768 online;
alter rollback segment T769 online;
alter rollback segment T770 online;
alter rollback segment T771 online;
alter rollback segment T772 online;
alter rollback segment T773 online;
alter rollback segment T774 online;
alter rollback segment T775 online;
alter rollback segment T776 online;
alter rollback segment T777 online;
alter rollback segment T778 online;
alter rollback segment T779 online;
alter rollback segment T780 online;
alter rollback segment T781 online;
alter rollback segment T782 online;
alter rollback segment T783 online;
alter rollback segment T784 online;
alter rollback segment T785 online;
alter rollback segment T786 online;
alter rollback segment T787 online;
alter rollback segment T788 online;
alter rollback segment T789 online;
alter rollback segment T790 online;
alter rollback segment T791 online;
alter rollback segment T792 online;
alter rollback segment T793 online;
alter rollback segment T794 online;
alter rollback segment T795 online;
alter rollback segment T796 online;
alter rollback segment T797 online;
alter rollback segment T798 online;
alter rollback segment T799 online;
alter rollback segment T800 online;

alter rollback segment S1 offline;
alter rollback segment S2 offline;
alter rollback segment S3 offline;
alter rollback segment S4 offline;
alter rollback segment S5 offline;
alter rollback segment S6 offline;
alter rollback segment S7 offline;
alter rollback segment S8 offline;
alter rollback segment S9 offline;
alter rollback segment S10 offline;
alter rollback segment S11 offline;
alter rollback segment S12 offline;
alter rollback segment S13 offline;
alter rollback segment S14 offline;
alter rollback segment S15 offline;
alter rollback segment S16 offline;
alter rollback segment S17 offline;
alter rollback segment S18 offline;
alter rollback segment S19 offline;
alter rollback segment S20 offline;
alter rollback segment S21 offline;
alter rollback segment S22 offline;
alter rollback segment S23 offline;
alter rollback segment S24 offline;
alter rollback segment S25 offline;
alter rollback segment S26 offline;
alter rollback segment S27 offline;
alter rollback segment S28 offline;
alter rollback segment S29 offline;
alter rollback segment S30 offline;
exit;
```

build/cust.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       tpcc_tab2.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop cluster ccluster including tables;
        drop table customer;

set timing on;
set echo on;

rem
rem CUSTOMER table
rem

        create cluster ccluster (
            c_id         number(5,0),
            c_d_id       number(2,0),
            c_w_id       number(5,0)
        )
          single              table
        hashkeys        224640000
        hash is         (c_id * (74880) + c_w_id*10 + c_d_id -1)
        size            850
        initrans        3
        pctfree         0
        tablespace      cust;

        create table customer (
        c_id            number(5,0),
        c_d_id          number(2,0),
        c_w_id          number(5,0),
        c_discount      number(4,4),
        c_credit        char(2),
        c_last          varchar2(16),
        c_first         varchar2(16),
        c_credit_lim    number(12),
        c_balance       number(12),
        c_ytd_payment   number(12),
        c_payment_cnt   number(8),
        c_delivery_cnt  number(8),
        c_street_1      varchar2(20),
        c_street_2      varchar2(20),
        c_city          varchar2(20),
        c_state         char(2),
        c_zip           char(9),
        c_phone         char(16),
        c_since         date,
        c_middle        char(2),
        c_data          varchar2(500)
        )
        cluster ccluster (c_id, c_d_id, c_w_id);

rem
rem done
rem

        exit;
```

build/dist.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       dist.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
    drop table district;
          drop cluster dcluster including tables;

set timing on
set echo on

rem
rem DISTRICT table
rem

          create cluster dcluster (
                    d_w_id    number(5,0),
                    d_id      number(2,0)
          )
          single              table
          hashkeys  74880
          hash is             (d_w_id - 1) * 10 + d_id
          size                1536
          initrans  3
          pctfree             0
        tablespace  system;

        create table district (
          d_id            number(2,0),
          d_w_id          number(5,0),
          d_ytd           number(12),
          d_tax           number(4,4),
          d_next_o_id     number,
          d_name          varchar2(10),
          d_street_1      varchar2(20),
          d_street_2      varchar2(20),
          d_city          varchar2(20),
          d_state         char(2),
          d_zip           char(9)
          )
          cluster dcluster (d_w_id, d_id);

rem
rem done
rem

        exit;
```

build/hist.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       hist.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem
```

```
rem
rem DROP all first
rem
        drop table history;

set timing on
set echo on

rem
rem HISTORY table
rem

       create table history (
        h_c_id          number,
        h_c_d_id        number,
        h_c_w_id        number,
        h_d_id          number,
        h_w_id          number,
        h_date          date,
        h_amount        number(6),
        h_data          varchar2(24)
        )
        tablespace hist
        initrans 4
        pctfree  5
        storage (freelist groups 43 freelists 19);

rem
rem done
rem

        exit;
```

build/icust.sql

```
rem
rem ================================================================+
rem         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
rem                    OPEN SYSTEMS PERFORMANCE GROUP                 |
rem                         All Rights Reserved                       |
rem ================================================================+
rem  FILENAME
rem       icust.sql
rem  DESCRIPTION
rem       Create customer index for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index icustomer;

set timing on
set echo on

rem
rem ICUST1 index
rem

       create unique index icustomer on customer(c_w_id, c_d_id, c_id)
        tablespace icust1
        initrans 3
          parallel 10
        storage (freelist groups 43 freelists 19)
        pctfree  1;

rem
rem done
rem

        exit;
```

build/icust2.sql

```
rem
rem ================================================================+
rem         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
rem                    OPEN SYSTEMS PERFORMANCE GROUP                 |
rem                         All Rights Reserved                       |
rem ================================================================+
rem  FILENAME
rem       icust2.sql
rem  DESCRIPTION
rem       Create customer index 2 for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index icustomer2;

set timing on
set echo on

rem
rem ICUST2 index
rem

       create unique index icustomer2 on customer(c_last, c_w_id, c_d_id, c_first)
        tablespace icust2
        initrans 3
          parallel 10
        storage (freelist groups 43 freelists 19)
        pctfree  1;

rem
rem done
rem

        exit;
```

build/idist.sql

```
rem
rem ================================================================+
rem         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
rem                    OPEN SYSTEMS PERFORMANCE GROUP                 |
rem                         All Rights Reserved                       |
rem ================================================================+
rem  FILENAME
rem       dist.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index idistrict;

set timing on
set echo on

rem
rem DISTRICT index
rem

       create unique index idistrict on district(d_w_id, d_id)
        tablespace system
        initrans 3
          parallel 5
        pctfree  5
        storage (freelist groups 43 freelists 19);

rem
```

```
rem done
rem

        exit;
```

build//iitem.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                     OPEN SYSTEMS PERFORMANCE GROUP                 |
rem |                           All Rights Reserved                      |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       item.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop index iitem;

set timing on
set echo on

rem
rem ITEM index
rem

       create unique index iitem on item(i_id)
        tablespace system
        initrans 4
          parallel 5
        pctfree  5
        storage (freelist groups 43 freelists 19);

rem
rem done
rem

        exit;
```

build//iordr.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                     OPEN SYSTEMS PERFORMANCE GROUP                 |
rem |                           All Rights Reserved                      |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       iordr.sql
rem  DESCRIPTION
rem       Create orders index for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop index iorders;

set timing on
set echo on

rem
rem ORDERS index
rem

       create unique index iorders on orders(o_w_id, o_d_id, o_id)
        tablespace iord1
        initrans 3
          parallel 10
        storage (freelist groups 43 freelists 19)
        pctfree  1;

rem
rem done
rem

        exit;
```

build/iordr2.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                     OPEN SYSTEMS PERFORMANCE GROUP                 |
rem |                           All Rights Reserved                      |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       iordr2.sql
rem  DESCRIPTION
rem       Create orders index 2 for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop index iorders2;

set timing on
set echo on

rem
rem ORDERS index 2
rem

       create unique index iorders2 on orders(o_c_id, o_d_id, o_w_id, o_id)
        tablespace iord2
        initrans 4
          parallel 10
        storage (freelist groups 43 freelists 19)
        pctfree  1;

rem
rem done
rem

        exit;
```

build/istok.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                     OPEN SYSTEMS PERFORMANCE GROUP                 |
rem |                           All Rights Reserved                      |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       istok.sql
rem  DESCRIPTION
rem       Create stock index for TPC-C database.
rem ====================================================================
rem

rem
rem DROP all first
rem
        drop index istock;
```

```
set timing on
set echo on

rem
rem STOCK index
rem

       create unique index istock on stock(s_i_id, s_w_id)
        tablespace istk
        initrans 3
          parallel 20
        storage (freelist groups 43 freelists 19)
        pctfree  1;

rem
rem done
rem

        exit;
```

build/item.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       item.sql
rem  DESCRIPTION
rem       Create ITEM table for TPC-C database.
rem ================================================================
rem

rem
rem DROP item cluster and table
rem
        drop cluster icluster including tables;
        drop table item;

set timing on;
set echo on;

rem
rem ITEM table
rem

        create cluster icluster (
        i_id            number(6,0)
        )
          single              table
        hashkeys        100000
        hash is         i_id
        size            120
        initrans        3
        pctfree         0
        tablespace      system;

        create table item (
        i_id            number(6,0),
        i_name          varchar2(24),
        i_price         number(5,0),
        i_data          varchar2(50),
        i_im_id         number
        )
        cluster icluster(i_id);


rem
rem done
rem

        exit;
```

build/iware.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       iware.sql
rem  DESCRIPTION
rem       Create warehouse index for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop index iwarehouse;

set timing on
set echo on

rem
rem WAREHOUSE index
rem

     create unique index iwarehouse on warehouse (w_id)
        tablespace system
        initrans 3
          pctfree 1;

rem
rem done
rem

        exit;
```

build/nord.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       nord.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop table new_order;

set timing on
set echo on

rem
rem NEW_ORDER table
rem

       create table new_order (
                no_w_id         number,
                no_d_id         number,
                no_o_id         number,
                    constraint inord primary key (no_w_id, no_d_id, no_o_id)
        )
          organization index
```

```
        tablespace nord
        initrans 4
        pctfree  5
        storage (freelist groups 43 freelists 19);

rem
rem done
rem

        exit;
```

build/ordl.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       ordr.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop table order_line;

set timing on
set echo on

rem
rem ORDER_LINE table
rem

       create table order_line (
        ol_w_id         number,
        ol_d_id         number,
        ol_o_id         number,
        ol_number       number,
        ol_i_id         number,
        ol_delivery_d   date,
        ol_amount       number(6),
        ol_supply_w_id  number,
        ol_quantity     number,
        ol_dist_info    char(24),
          constraint iordl primary key (ol_w_id, ol_d_id, ol_o_id, ol_number)
        )
          organization index
        tablespace ordl
        initrans 4
        pctfree  5
        storage (freelist groups 43 freelists 19);

rem
rem done
rem

        exit;
```

build/ordr.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       ordr.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop table orders;

set timing on
set echo on

rem
rem ORDERS table
rem

       create table orders (
        o_id            number,
        o_w_id          number,
        o_d_id          number,
        o_c_id          number,
        o_carrier_id    number,
        o_ol_cnt        number,
        o_all_local     number,
        o_entry_d       date
        )
        tablespace ordr
        initrans 4
        pctfree  5
        storage (freelist groups 43 freelists 19);

rem
rem done
rem

        exit;
```

build/stok.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                    OPEN SYSTEMS PERFORMANCE GROUP                  |
rem |                         All Rights Reserved                        |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       tpcc_tab3.sql
rem  DESCRIPTION
rem       Create stock table for TPC-C database.
rem ================================================================
rem

rem
rem DROP all first
rem
        drop cluster scluster including tables;
        drop table stock;

set timing on;
set echo on;

rem
rem STOCK table
rem

        create cluster scluster (
           s_i_id       number(6,0),
           s_w_id       number(5,0)
        )
          single              table
        hashkeys        748800000
        hash is         s_i_id * 7488 + s_w_id
        size            350
        initrans        3
```

```
        pctfree         0
        tablespace      stok
        storage (freelist groups 43 freelists 19);

        create table stock (
        s_i_id          number(6,0),
        s_w_id          number(5,0),
        s_quantity      number(6,0),
        s_ytd           number(10,0),
        s_order_cnt     number(6,0),
        s_remote_cnt    number(6,0),
        s_data          varchar2(50),
        s_dist_01       char(24),
        s_dist_02       char(24),
        s_dist_03       char(24),
        s_dist_04       char(24),
        s_dist_05       char(24),
        s_dist_06       char(24),
        s_dist_07       char(24),
        s_dist_08       char(24),
        s_dist_09       char(24),
        s_dist_10       char(24)
        )
        cluster scluster (s_i_id, s_w_id);

rem
rem done
rem

        exit;
```

build/ware.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
rem |                   OPEN SYSTEMS PERFORMANCE GROUP                   |
rem |                      All Rights Reserved                           |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       ware.sql
rem  DESCRIPTION
rem       Create customer table for TPC-C database.
rem ====================================================================
rem


rem
rem DROP all first
rem
        drop table warehouse;
          drop cluster wcluster including tables;

set timing on
set echo on

rem
rem WAREHOUSE table
rem

          create cluster wcluster (
                    w_id                number(5,0)
          )
          single                        table
          hashkeys              7488
          hash is                       w_id
          size                          1536
          initrans              3
          pctfree                       0
        tablespace  system;

        create table warehouse (
                w_id            number(5,0),
                w_ytd           number(12),
                w_tax           number(4,4),
                w_name          varchar2(10),
                w_street_1      varchar2(20),
                w_street_2      varchar2(20),
                w_city          varchar2(20),
                w_state         char(2),
                w_zip           char(9)
        )
          cluster wcluster (w_id);

rem
rem done
rem

        exit;
```

build/sql/tpcc_ana.sql

```
rem
rem +--------------------------------------------------------------------+
rem |         Copyright (c) 1995  Oracle Corp, Redwood Shores, CA        |
rem |                   OPEN SYSTEMS PERFORMANCE GROUP                   |
rem |                      All Rights Reserved                           |
rem +--------------------------------------------------------------------+
rem  FILENAME
rem       tpcc_ana.sql
rem  DESCRIPTION
rem       Analyze all tables and indexes of TPC-C database.
rem ====================================================================
rem

set timing on;
set echo on;
analyze table warehouse compute statistics;
analyze table district compute statistics;
analyze table item estimate statistics;
analyze table history estimate statistics;
analyze table customer estimate statistics;
analyze table stock estimate statistics;
analyze table orders estimate statistics;
rem analyze table new_order estimate statistics;
rem analyze table order_line estimate statistics;
analyze cluster wcluster compute statistics;
analyze cluster dcluster compute statistics;
analyze cluster icluster estimate statistics;
analyze cluster scluster estimate statistics;
analyze cluster ccluster estimate statistics;
analyze index iwarehouse compute statistics;
analyze index idistrict compute statistics;
analyze index icustomer estimate statistics;
analyze index icustomer2 estimate statistics;
analyze index istock estimate statistics;
analyze index iitem estimate statistics;
analyze index iorders estimate statistics;
analyze index iorders2 estimate statistics;
analyze index inord estimate statistics;
analyze index iordl estimate statistics;
quit;
```

10.2 Loader Program

build/loader/tpccload.c

```
#ifdef RCSID
static char *RCSid =
```

```
   "$Header: tpccload.c 7030100.1 96/05/13 16:20:36 plai Generic<base> $ Copyr (c)
1993 Oracle";
#endif /* RCSID */

/*+=================================================================+
 |        Copyright (c) 1994  Oracle Corp, Redwood Shores, CA      |
 |                    OPEN SYSTEMS PERFORMANCE GROUP               |
 |                         All Rights Reserved                     |
 +=================================================================+
 | FILENAME
 |    tpccload.c
 | DESCRIPTION
 |    Load or generate TPC-C database tables.
 |    Usage: tpccload -M <# of warehouses> [options]
 |              options: -A  load all tables
 |                       -w  load warehouse table
 |                       -d  load district table
 |                       -c  load customer table
 |                       -i  load item table
 |                       -s  load stock table (cluster around s_w_id)
 |                       -S  load stock table (cluster around s_i_id)
 |                       -h  load history table
 |                       -n  load new-order table
 |                       -o <oline file> load order and order-line table
 |                       -b <ware#>  beginning warehouse number
 |                       -e <ware#>  ending warehouse number
 |                       -j <item#>  beginning item number (with -S)
 |                       -k <item#>  ending item number (with -S)
 |                       -g  generate rows to standard output
 +=================================================================*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <sys/types.h>
#include "tpcc.h"

#define DISTARR 10            /* district insert array size */
#define CUSTARR 100           /* customer insert array size */
#define STOCARR 100           /* stock insert array size    */
#define ITEMARR 100           /* item insert array size     */
#define HISTARR 100             /* history insert array size    */
#define ORDEARR 100             /* order insert array size      */
#define NEWOARR 100             /* new order insert array size  */

#define DISTFAC 10            /* max. disctrict id          */
#define CUSTFAC 3000                    /* max. customer id */
#define STOCFAC 100000                  /* max. stock id    */
#define ITEMFAC 100000                  /* max. item id               */
#define HISTFAC 30000           /* history / warehouse  */
#define ORDEFAC 3000            /* order / district     */
#define NEWOFAC 900             /* new order / district */

#define C       0               /* constant in non-uniform dist. eqt. */
#define CNUM1   1               /* first constant in non-uniform dist. eqt. */
#define CNUM2   2               /* second constant in non-uniform dist. eqt. */
#define CNUM3   3               /* third constant in non-uniform dist. eqt. */

#define SEED    2               /* seed for random functions */

#define SQLTXTW "INSERT INTO warehouse (w_id, w_ytd, w_tax, w_name, w_street_1,
w_street_2, w_city, w_state, w_zip) VALUES (:w_id, 30000000, :w_tax, :w_name,
:w_street_1, \
   :w_street_2, :w_city, :w_state, :w_zip)"

#define SQLTXTD "INSERT INTO district (d_id, d_w_id, d_ytd, d_tax, d_next_o_id,
d_name, d_street_1, d_street_2, d_city, d_state, d_zip) VALUES (:d_id,
:d_w_id,3000000, :d_tax, \
     3001, :d_name, :d_street_1, :d_street_2, :d_city, :d_state, :d_zip)"

#define SQLTXTC "INSERT INTO customer (C_ID, C_D_ID, C_W_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, C_DELIVERY_CNT,
C_DATA) VALUES (:c_id, :c_d_id, :c_w_id, \
   :c_first, 'OE', :c_last, :c_street_1, :c_street_2, :c_city, :c_state, \
   :c_zip, :c_phone, SYSDATE, :c_credit, 5000000, :c_discount, -1000, 1000, 1, \
   0, :c_data)"

#define SQLTXTH "INSERT INTO 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 (:h_c_id, :h_c_d_id, :h_c_w_id, \
   :h_d_id, :h_w_id, SYSDATE, 1000, :h_data)"

#define SQLTXTS "INSERT INTO stock (s_i_id, s_w_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) \
VALUES (:s_i_id, :s_w_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, 0, 0, 0, :s_data)" \

#define SQLTXTI "INSERT INTO item (I_ID,I_IM_ID,I_NAME,I_PRICE,I_DATA) VALUES
(:i_id, :i_im_id, :i_name, :i_price, \
   :i_data)"

#define SQLTXTO1 "INSERT INTO orders (O_ID,
O_D_ID,O_W_ID,O_C_ID,O_ENTRY_D,O_CARRIER_ID,O_OL_CNT,O_ALL_LOCAL) \
   VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
   SYSDATE, :o_carrier_id, :o_ol_cnt, 1)"

#define SQLTXTO2 "INSERT INTO orders (O_ID,
O_D_ID,O_W_ID,O_C_ID,O_ENTRY_D,O_CARRIER_ID,O_OL_CNT,O_ALL_LOCAL) \
   VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
   SYSDATE, 11, :o_ol_cnt, 1)"

#define SQLTXTOL1 "INSERT INTO order_line (OL_O_ID, OL_D_ID, OL_W_ID, OL_NUMBER,
OL_DELIVERY_D, OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY, OL_AMOUNT, OL_DIST_INFO) \
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number, SYSDATE, :ol_i_id, :ol_supply_w_id, 5, 0, \
   :ol_dist_info)"

#define SQLTXTOL2 "INSERT INTO order_line (OL_O_ID, OL_D_ID, OL_W_ID, OL_NUMBER,
OL_DELIVERY_D, OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY, OL_AMOUNT, OL_DIST_INFO) \
   VALUES (:ol_o_id, :ol_d_id, \
   :ol_w_id, :ol_number, to_date('01-Jan-1811'), :ol_i_id, :ol_supply_w_id, 5,
:ol_amount, \
   :ol_dist_info)"

#define SQLTXTNO "INSERT INTO new_order (no_o_id, no_d_id, no_w_id) VALUES
(:no_o_id, :no_d_id, :no_w_id)"

ldadef tpclda;
csrdef curw, curd, curc, curh, curs, curi, curo1, curo2, curol1, curol2, curno;
unsigned long tpchda[256];

static char *lastname[] = {
   "BAR",
   "OUGHT",
   "ABLE",
   "PRI",
   "PRES",
   "ESE",
   "ANTI",
   "CALLY",
   "ATION",
   "EING"
};

char num9[10];
char num16[17];
char str2[3];
char str24[15][25];
int randperm3000[3000];



myusage()

{

   fprintf (stderr, "\n");
   fprintf (stderr, "Usage:\ttpccload -M <multiplier> [options]\n");
   fprintf (stderr, "options:\n");
   fprintf (stderr, "\t-A :\tload all tables\n");
   fprintf (stderr, "\t-w :\tload warehouse table\n");
   fprintf (stderr, "\t-d :\tload district table\n");
   fprintf (stderr, "\t-c :\tload customer table\n");
   fprintf (stderr, "\t-i :\tload item table\n");
```

```
   fprintf (stderr, "\t-s :\tload stock table (cluster around s_w_id)\n");
   fprintf (stderr, "\t-S :\tload stock table (cluster around s_i_id)\n");
   fprintf (stderr, "\t-h :\tload history table\n");
   fprintf (stderr, "\t-n :\tload new-order table\n");
   fprintf (stderr, "\t-o <oline file> :\tload order and order-line table\n");
   fprintf (stderr, "\t-b <ware#> :\tbeginning warehouse number\n");
   fprintf (stderr, "\t-e <ware#> :\tending warehouse number\n");
   fprintf (stderr, "\t-j <item#> :\tbeginning item number (with -S)\n");
   fprintf (stderr, "\t-k <item#> :\tending item number (with -S)\n");
   fprintf (stderr, "\t-g :\tgenerate rows to standard output\n");
   fprintf (stderr, "\n");
   exit(1);

}



errrpt (lda, cur)

csrdef *lda;
csrdef *cur;

{

   text msg[2048];

   if (cur->rc) {
      oerhms (lda, cur->rc, msg, 2048);
      fprintf (stderr, "TPC-C load error: %s\n", msg);
   }

}



quit ()

{

   if (oclose (&curw))
      errrpt (&tpclda, &curw);

   if (oclose (&curd))
      errrpt (&tpclda, &curd);

   if (oclose (&curc))
      errrpt (&tpclda, &curc);

   if (oclose (&curh))
      errrpt (&tpclda, &curh);

   if (oclose (&curs))
      errrpt (&tpclda, &curs);

   if (oclose (&curi))
      errrpt (&tpclda, &curi);

   if (oclose (&curo1))
      errrpt (&tpclda, &curo1);

   if (oclose (&curo2))
      errrpt (&tpclda, &curo2);

   if (oclose (&curol1))
      errrpt (&tpclda, &curol1);

   if (oclose (&curol2))
      errrpt (&tpclda, &curol2);

   if (oclose (&curno))
      errrpt (&tpclda, &curno);

   if (ologof (&tpclda))
      fprintf (stderr, "TPC-C load error: Error in logging off\n");

}
```

```
main (argc, argv)

int argc;
char *argv[];

{

   char *uid="tpcc/tpcc";
   text sqlbuf[1024];
   int scale=0;
   int i, j;
   int loop;
   int loopcount;
   int cid;
   int dwid;
   int cdid;
   int cwid;
   int sid;
   int swid;
   int olcnt;
   int nrows;
   int row;

   int w_id;
   char w_name[11];
   char w_street_1[21];
   char w_street_2[21];
   char w_city[21];
   char w_state[2];
   char w_zip[9];
   float w_tax;

   int d_id[10];
   int d_w_id[10];
   char d_name[10][11];
   char d_street_1[10][21];
   char d_street_2[10][21];
   char d_city[10][21];
   char d_state[10][2];
   char d_zip[10][9];
   float d_tax[10];

   int c_id[100];
   int c_d_id[100];
   int c_w_id[100];
   char c_first[100][17];
   char c_last[100][17];
   char c_street_1[100][21];
   char c_street_2[100][21];
   char c_city[100][21];
   char c_state[100][2];
   char c_zip[100][9];
   char c_phone[100][16];
   char c_credit[100][2];
   float c_discount[100];
   char c_data[100][501];

   int i_id[100];
   int i_im_id[100];
   int i_price[100];
   char i_name[100][25];
   char i_data[100][51];

   int s_i_id[100];
   int s_w_id[100];
   int s_quantity[100];
   char s_dist_01[100][24];
   char s_dist_02[100][24];
   char s_dist_03[100][24];
   char s_dist_04[100][24];
   char s_dist_05[100][24];
   char s_dist_06[100][24];
   char s_dist_07[100][24];
   char s_dist_08[100][24];
   char s_dist_09[100][24];
   char s_dist_10[100][24];
   char s_data[100][51];

   int h_w_id[100];
```

```
   int h_d_id[100];
   int h_c_id[100];
   char h_data[100][25];

   int o_id[100];
   int o_d_id[100];
   int o_w_id[100];
   int o_c_id[100];
   int o_carrier_id[100];
   int o_ol_cnt[100];

   int ol_o_id[15];
   int ol_d_id[15];
   int ol_w_id[15];
   int ol_number[15];
   int ol_i_id[15];
   int ol_supply_w_id[15];
   int ol_amount[15];
   char ol_dist_info[15][24];

   int no_o_id[100];
   int no_d_id[100];
   int no_w_id[100];

   char sdate[30];

   double begin_time, end_time;
   double begin_cpu, end_cpu;
   double gettime(), getcpu();

   extern int getopt();
   extern char *optarg;
   extern int optind, opterr;

   char   *argstr="M:AwdcisShno:b:e:j:k:g";
   int opt;
   int do_A=0;
   int do_w=0;
   int do_d=0;
   int do_i=0;
   int do_c=0;
   int do_s=0;
   int do_S=0;
   int do_h=0;
   int do_o=0;
   int do_n=0;
   int gen=0;
   int bware=1;
   int eware=0;
   int bitem=1;
   int eitem=0;

   FILE *olfp=NULL;
   char olfname[100];

/*-------------------------------------------------------------+
 | Parse command line -- look for scale factor.                |
 +-------------------------------------------------------------*/

   if (argc == 1) {
      myusage ();
   }

   while ((opt = getopt (argc, argv, argstr)) != -1) {
      switch (opt) {
         case '?': myusage ();
                   break;
         case 'M': scale = atoi (optarg);
                   break;
         case 'A': do_A = 1;
                   break;
         case 'w': do_w = 1;
                   break;
         case 'd': do_d = 1;
                   break;
         case 'c': do_c = 1;
                   break;
         case 'i': do_i = 1;
                   break;
         case 's': do_s = 1;
                   break;
         case 'S': do_S = 1;
                   break;
         case 'h': do_h = 1;
                   break;
         case 'n': do_n = 1;
                   break;
         case 'o': do_o = 1;
                   strcpy (olfname, optarg);
                   break;
         case 'b': bware = atoi (optarg);
                   break;
         case 'e': eware = atoi (optarg);
                   break;
         case 'j': bitem = atoi (optarg);
                   break;
         case 'k': eitem = atoi (optarg);
                   break;
         case 'g': gen = 1;
                   break;
         default: fprintf (stderr, "THIS SHOULD NEVER HAPPEN!!!\n");
                  fprintf (stderr, "(reached default case in getopt ())\n");
                  myusage ();
      }
   }

/*-------------------------------------------------------------*|
|         Rudimentary error checking                           |
|*-------------------------------------------------------------*/

   if (scale  < 1) {
      fprintf (stderr, "Invalid scale factor: '%d'\n", scale);
      myusage ();
   }

   if (!(do_A || do_w || do_d || do_c || do_i || do_s || do_S || do_h || do_o ||
         do_n)) {
      fprintf (stderr, "What should I load???\n");
      myusage ();
   }

   if (gen && (do_A || (do_w + do_d + do_c + do_i + do_s + do_S + do_h + do_o +
                        do_n > 1))) {
      fprintf (stderr, "Can only generate table one at a time\n");
      myusage ();
   }

   if (do_S && (do_A || do_s)) {
      fprintf (stderr, "Cluster stock table around s_w_id or s_i_id?\n");
      myusage ();
   }

   if (eware <= 0)
      eware = scale;
   if (eitem <= 0)
      eitem = STOCFAC;

   if (do_S) {
      if ((bitem  < 1) || (bitem > STOCFAC)) {
         fprintf (stderr, "Invalid beginning item number: '%d'\n", bitem);
         myusage ();
      }

      if ((eitem  < bitem) || (eitem > STOCFAC)) {
         fprintf (stderr, "Invalid ending item number: '%d'\n", eitem);
         myusage ();
      }
   }

   if ((bware  < 1) || (bware > scale)) {
      fprintf (stderr, "Invalid beginning warehouse number: '%d'\n", bware);
      myusage ();
   }

   if ((eware  < bware) || (eware > scale)) {
      fprintf (stderr, "Invalid ending warehouse number: '%d'\n", eware);
      myusage ();
   }
```

```
   if (gen && do_o) {
      if ((olfp = fopen (olfname, "w")) == NULL) {
         fprintf (stderr, "Can't open '%s' for writing order lines\n", olfname);
         myusage ();
      }
   }

/*---------------------------------------------------------------+
 | Prepare to insert into database.                              |
 +---------------------------------------------------------------*/

   sysdate (sdate);
   if (!gen) {

      /* log on to Oracle */

      if (orlon (&tpclda, (ub1 *) tpchda, (text *) uid, -1, (text *) 0, -1, 0)) {
         fprintf (stderr, "TPC-C load error: Error in logging on\n");
         errrpt (&tpclda, &tpclda);
         exit (1);
      }

      fprintf (stderr, "\nConnected to Oracle userid '%s'.\n", uid);

      /* turn off auto-commit */

      if (ocof (&tpclda)) {
         errrpt (&tpclda, &tpclda);
         ologof (&tpclda);
         exit (1);
      }

      /* open cursors */

      if (oopen (&curw, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curw);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curd, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curd);
         oclose (&curw);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curc, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curc);
         oclose (&curw);
         oclose (&curd);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curh, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curh);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curs, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curs);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curi, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curi);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         oclose (&curs);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curo1, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curo1);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         oclose (&curs);
         oclose (&curi);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curo2, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curo2);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         oclose (&curs);
         oclose (&curi);
         oclose (&curo1);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curol1, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curol1);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         oclose (&curs);
         oclose (&curi);
         oclose (&curo1);
         oclose (&curo2);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curol2, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curol2);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         oclose (&curs);
         oclose (&curi);
         oclose (&curo1);
         oclose (&curo2);
         oclose (&curol1);
         ologof (&tpclda);
         exit (1);
      }

      if (oopen (&curno, &tpclda, (text *) 0, -1, -1, (text *) uid, -1)) {
         errrpt (&tpclda, &curno);
         oclose (&curw);
         oclose (&curd);
         oclose (&curc);
         oclose (&curh);
         oclose (&curs);
         oclose (&curi);
         oclose (&curo1);
         oclose (&curo2);
         oclose (&curol1);
         oclose (&curol2);
         ologof (&tpclda);
         exit (1);
      }

      /* parse statements */
```

```
sprintf ((char *) sqlbuf, SQLTXTW);
if (oparse (&curw, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTD);
if (oparse (&curd, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTC);
if (oparse (&curc, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTH);
if (oparse (&curh, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTS);
if (oparse (&curs, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTI);
if (oparse (&curi, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curi);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTO1);
if (oparse (&curo1, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTO2);
if (oparse (&curo2, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curo2);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTOL1);
if (oparse (&curol1, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTOL2);
if (oparse (&curol2, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

sprintf ((char *) sqlbuf, SQLTXTNO);
if (oparse (&curno, sqlbuf, -1, 0, 1)) {
   errrpt (&tpclda, &curno);
   quit ();
   exit (1);
}

/* bind variables */

/* warehouse */

if (obndrv (&curw, (text *) ":w_id", -1, (ub1 *) &w_id, sizeof (w_id),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_name", -1, (ub1 *) w_name, 11,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_street_1", -1, (ub1 *) w_street_1, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_street_2", -1, (ub1 *) w_street_2, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_city", -1, (ub1 *) w_city, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_state", -1, (ub1 *) w_state, 2,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_zip", -1, (ub1 *) w_zip, 9,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

if (obndrv (&curw, (text *) ":w_tax", -1, (ub1 *) &w_tax, sizeof (w_tax),
          SQLT_FLT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curw);
   quit ();
   exit (1);
}

/* district */

if (obndrv (&curd, (text *) ":d_id", -1, (ub1 *) d_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_w_id", -1, (ub1 *) d_w_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_name", -1, (ub1 *) d_name, 11,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
```

```
}

if (obndrv (&curd, (text *) ":d_street_1", -1, (ub1 *) d_street_1, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_street_2", -1, (ub1 *) d_street_2, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_city", -1, (ub1 *) d_city, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_state", -1, (ub1 *) d_state, 2,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_zip", -1, (ub1 *) d_zip, 9,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

if (obndrv (&curd, (text *) ":d_tax", -1, (ub1 *) d_tax, sizeof (int),
          SQLT_FLT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curd);
   quit ();
   exit (1);
}

/* customer */

if (obndrv (&curc, (text *) ":c_id", -1, (ub1 *) c_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_d_id", -1, (ub1 *) c_d_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_w_id", -1, (ub1 *) c_w_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_first", -1, (ub1 *) c_first, 17,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_last", -1, (ub1 *) c_last, 17,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_street_1", -1, (ub1 *) c_street_1, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_street_2", -1, (ub1 *) c_street_2, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_city", -1, (ub1 *) c_city, 21,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_state", -1, (ub1 *) c_state, 2,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_zip", -1, (ub1 *) c_zip, 9,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_phone", -1, (ub1 *) c_phone, 16,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_credit", -1, (ub1 *) c_credit, 2,
          SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_discount", -1, (ub1 *) c_discount,
          sizeof (int), SQLT_FLT, -1, (sb2 *) 0, (text *) 0, -1,
          -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

if (obndrv (&curc, (text *) ":c_data", -1, (ub1 *) c_data, 501,
          SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curc);
   quit ();
   exit (1);
}

/* item */

if (obndrv (&curi, (text *) ":i_id", -1, (ub1 *) i_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curi);
   quit ();
   exit (1);
}

if (obndrv (&curi, (text *) ":i_im_id", -1, (ub1 *) i_im_id, sizeof (int),
          SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curi);
   quit ();
```

```
   exit (1);
}

if (obndrv (&curi, (text *) ":i_name", -1, (ub1 *) i_name, 25,
            SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curi);
   quit ();
   exit (1);
}

if (obndrv (&curi, (text *) ":i_price", -1, (ub1 *) i_price,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1,
            -1)) {
   errrpt (&tpclda, &curi);
   quit ();
   exit (1);
}

if (obndrv (&curi, (text *) ":i_data", -1, (ub1 *) i_data, 51,
            SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curi);
   quit ();
   exit (1);
}

/* stock */

if (obndrv (&curs, (text *) ":s_i_id", -1, (ub1 *) s_i_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_w_id", -1, (ub1 *) s_w_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_quantity", -1, (ub1 *) s_quantity,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_01", -1, (ub1 *) s_dist_01, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_02", -1, (ub1 *) s_dist_02, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_03", -1, (ub1 *) s_dist_03, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_04", -1, (ub1 *) s_dist_04, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_05", -1, (ub1 *) s_dist_05, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_06", -1, (ub1 *) s_dist_06, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_07", -1, (ub1 *) s_dist_07, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_08", -1, (ub1 *) s_dist_08, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_09", -1, (ub1 *) s_dist_09, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_dist_10", -1, (ub1 *) s_dist_10, 24,
            SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

if (obndrv (&curs, (text *) ":s_data", -1, (ub1 *) s_data, 51,
            SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curs);
   quit ();
   exit (1);
}

/* history */

if (obndrv (&curh, (text *) ":h_c_id", -1, (ub1 *) h_c_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

if (obndrv (&curh, (text *) ":h_c_d_id", -1, (ub1 *) h_d_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

if (obndrv (&curh, (text *) ":h_c_w_id", -1, (ub1 *) h_w_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

if (obndrv (&curh, (text *) ":h_d_id", -1, (ub1 *) h_d_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

if (obndrv (&curh, (text *) ":h_w_id", -1, (ub1 *) h_w_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curh);
```

```
   quit ();
   exit (1);
}

if (obndrv (&curh, (text *) ":h_data", -1, (ub1 *) h_data, 25,
            SQLT_STR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curh);
   quit ();
   exit (1);
}

/* order_line (delivered) */

if (obndrv (&curol1, (text *) ":ol_o_id", -1, (ub1 *) ol_o_id,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

if (obndrv (&curol1, (text *) ":ol_d_id", -1, (ub1 *) ol_d_id,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

if (obndrv (&curol1, (text *) ":ol_w_id", -1, (ub1 *) ol_w_id,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

if (obndrv (&curol1, (text *) ":ol_number", -1, (ub1 *) ol_number,
    sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

if (obndrv (&curol1, (text *) ":ol_i_id", -1, (ub1 *) ol_i_id,
    sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

if (obndrv (&curol1, (text *) ":ol_supply_w_id", -1,
            (ub1 *) ol_supply_w_id, sizeof (int), SQLT_INT, -1,
            (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

if (obndrv (&curol1, (text *) ":ol_dist_info", -1, (ub1 *) ol_dist_info,
            24, SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol1);
   quit ();
   exit (1);
}

/* order_line (not delivered) */

if (obndrv (&curol2, (text *) ":ol_o_id", -1, (ub1 *) ol_o_id,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_d_id", -1, (ub1 *) ol_d_id,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_w_id", -1, (ub1 *) ol_w_id,
            sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_number", -1, (ub1 *) ol_number,
    sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_i_id", -1, (ub1 *) ol_i_id,
    sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_supply_w_id", -1,
            (ub1 *) ol_supply_w_id, sizeof (int), SQLT_INT, -1,
            (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_amount", -1, (ub1 *) ol_amount,
    sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

if (obndrv (&curol2, (text *) ":ol_dist_info", -1, (ub1 *) ol_dist_info,
            24, SQLT_CHR, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curol2);
   quit ();
   exit (1);
}

/* orders (delivered) */

if (obndrv (&curo1, (text *) ":o_id", -1, (ub1 *) o_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}

if (obndrv (&curo1, (text *) ":o_d_id", -1, (ub1 *) o_d_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}

if (obndrv (&curo1, (text *) ":o_w_id", -1, (ub1 *) o_w_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}

if (obndrv (&curo1, (text *) ":o_c_id", -1, (ub1 *) o_c_id, sizeof (int),
            SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}

if (obndrv (&curo1, (text *) ":o_carrier_id", -1, (ub1 *) o_carrier_id,
    sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
   errrpt (&tpclda, &curo1);
   quit ();
   exit (1);
}
```

```
      if (obndrv (&curo1, (text *) ":o_ol_cnt", -1, (ub1 *) o_ol_cnt,
          sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo1);
         quit ();
         exit (1);
      }

      /* orders (not delivered) */

      if (obndrv (&curo2, (text *) ":o_id", -1, (ub1 *) o_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_d_id", -1, (ub1 *) o_d_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_w_id", -1, (ub1 *) o_w_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_c_id", -1, (ub1 *) o_c_id, sizeof (int),
                  SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      if (obndrv (&curo2, (text *) ":o_ol_cnt", -1, (ub1 *) o_ol_cnt,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curo2);
         quit ();
         exit (1);
      }

      /* new order */

      if (obndrv (&curno, (text *) ":no_o_id", -1, (ub1 *) no_o_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curno);
         quit ();
         exit (1);
      }

      if (obndrv (&curno, (text *) ":no_d_id", -1, (ub1 *) no_d_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curno);
         quit ();
         exit (1);
      }

      if (obndrv (&curno, (text *) ":no_w_id", -1, (ub1 *) no_w_id,
                  sizeof (int), SQLT_INT, -1, (sb2 *) 0, (text *) 0, -1, -1)) {
         errrpt (&tpclda, &curno);
         quit ();
         exit (1);
      }
   }

/*---------------------------------------------------------------+
 | Initialize random number generator                            |
 +---------------------------------------------------------------*/

   srand (SEED);
   srand48 (SEED);
   initperm ();


/*---------------------------------------------------------------+
 | Load the WAREHOUSE table.                                     |
 +---------------------------------------------------------------*/

   if (do_A || do_w) {
      nrows = eware - bware + 1;

      fprintf (stderr, "Loading/generating warehouse: w%d - w%d (%d rows)\n",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      for (loop = bware; loop <= eware; loop++) {

         w_tax = (rand () % 2001) * 0.0001;
         randstr (w_name, 6, 10);
         randstr (w_street_1, 10, 20);
         randstr (w_street_2, 10, 20);
         randstr (w_city, 10, 20);
         randstr (str2, 2, 2);
         randnum (num9, 9);
         num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';

         if (gen) {
            printf ("%d 30000000 %6.4f %s %s %s %s %s %s\n", loop, w_tax,
                    w_name, w_street_1, w_street_2, w_city, str2, num9);
            fflush (stdout);
         }
         else {
            w_id = loop;
            strncpy (w_state, str2, 2);
            strncpy (w_zip, num9, 9);

            if (oexec (&curw)) {
               errrpt (&tpclda, &curw);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d\n", loop);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d\n", loop);
               quit ();
               exit (1);
            }
         }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*---------------------------------------------------------------+
 | Load the DISTRICT table.                                      |
 +---------------------------------------------------------------*/

   if (do_A || do_d) {
      nrows = (eware - bware + 1) * DISTFAC;

      fprintf (stderr, "Loading/generating district: w%d - w%d (%d rows)\n",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      dwid = bware - 1;

      for (row = 0; row < nrows; ) {
         dwid++;

         for (i = 0; i < DISTARR; i++, row++) {
            d_tax[i] = (rand () % 2001) * 0.0001;
            randstr (d_name[i], 6, 10);
            randstr (d_street_1[i], 10, 20);
```

```
            randstr (d_street_2[i], 10, 20);
            randstr (d_city[i], 10, 20);
            randstr (str2, 2, 2);
            randnum (num9, 9);
            num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';

            if (gen) {
               /* printf ("%d %d %s %s %s %s %s %s %d 30000.0 3001\n",
                       i + 1, dwid, d_name[i], d_street_1[i], d_street_2[i],
                       d_city[i], str2, num9, d_tax[i]); */
                          /* Reordered columns */
               printf ("%d %d 3000000 %6.4f 3001 %s %s %s %s %s %s\n",
                       i + 1, dwid, d_tax[i], d_name[i], d_street_1[i],
                           d_street_2[i], d_city[i], str2, num9 );
            }
            else {
               d_id[i] = i + 1;
               d_w_id[i] = dwid;
               strncpy (d_state[i], str2, 2);
               strncpy (d_zip[i], num9, 9);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curd, DISTARR, 0)) {
               errrpt (&tpclda, &curd);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d, district 1\n", dwid);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at warehouse %d, district 1\n", dwid);
               quit ();
               exit (1);
            }
         }
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------------+
 | Load the CUSTOMER table.                                          |
 +-------------------------------------------------------------------*/

   if (do_A || do_c) {
      nrows = (eware - bware + 1) * CUSTFAC * DISTFAC;

      fprintf (stderr, "Loading/generating customer: w%d - w%d (%d rows)\n   ",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < CUSTARR; i++, row++) {
            cid++;
            if (cid > CUSTFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift district cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift warehouse cycle */
               }
            }
            c_id[i] = cid;
            c_d_id[i] = cdid;
            c_w_id[i] = cwid;
            if (cid <= 1000)
               randlastname (c_last[i], cid - 1);
            else
               randlastname (c_last[i], NURand (255, 0, 999, CNUM1));
            c_credit[i][1] = 'C';
            if (rand () % 10)
               c_credit[i][0] = 'G';
            else
               c_credit[i][0] = 'B';
            c_discount[i] = (rand () % 5001) * 0.0001;
            randstr (c_first[i], 8, 16);
            randstr (c_street_1[i], 10, 20);
            randstr (c_street_2[i], 10, 20);
            randstr (c_city[i], 10, 20);
            randstr (str2, 2, 2);
            randnum (num9, 9);
            num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';
            randnum (num16, 16);
            randstr (c_data[i], 300, 500);

            if (gen) {
               printf ("%d %d %d %s OE %s %s %s %s %s %s %s %s %cC 5000000 %6.4f -
1000 1000 1 0 %s\n",
                       cid, cdid, cwid, c_first[i], c_last[i],
                       c_street_1[i], c_street_2[i], c_city[i], str2, num9,
                       num16, sdate, c_credit[i][0], c_discount[i], c_data[i]);
            }
            else {
               strncpy (c_state[i], str2, 2);
               strncpy (c_zip[i], num9, 9);
               strncpy (c_phone[i], num16, 16);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curc, CUSTARR, 0)) {
               errrpt (&tpclda, &curc);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        c_w_id[0], c_d_id[0], c_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        c_w_id[0], c_d_id[0], c_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }


/*-------------------------------------------------------------------+
 | Load the ITEM table.                                              |
 +-------------------------------------------------------------------*/

   if (do_A || do_i) {
```

```
      nrows = ITEMFAC;

      fprintf (stderr, "Loading/generating item: (%d rows)\n   ", nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < ITEMARR; i++, row++) {
            i_im_id[i] = (rand () % 10000) + 1;
            i_price[i] = ((rand () % 9901) + 100);
            randstr (i_name[i], 14, 24);
            randdatastr (i_data[i], 26, 50);

            if (gen) {
               printf ("%d %d %s %d %s\n", row + 1, i_im_id[i], i_name[i],
                       i_price[i], i_data[i]);
            }
            else {
               i_id[i] = row + 1;
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curi, ITEMARR, 0)) {
               errrpt (&tpclda, &curi);
               orol (&tpclda);
               fprintf (stderr, "Aborted at i_id %d\n", i_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at i_id %d\n", i_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }


/*-----------------------------------------------------------------+
 | Load the STOCK table.                                           |
 +-----------------------------------------------------------------*/

   if (do_A || do_s) {

      nrows = (eware - bware + 1) * STOCFAC;

      fprintf (stderr, "Loading/generating stock: w%d - w%d (%d rows)\n   ",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      sid = 0;
      swid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < STOCARR; i++, row++) {
            if (++sid > STOCFAC) {        /* cheap mod */
               sid = 1;
               swid++;
            }
            s_quantity[i] = (rand () % 91) + 10;
            randstr (str24[0], 24, 24);
            randstr (str24[1], 24, 24);
            randstr (str24[2], 24, 24);
            randstr (str24[3], 24, 24);
            randstr (str24[4], 24, 24);
            randstr (str24[5], 24, 24);
            randstr (str24[6], 24, 24);
            randstr (str24[7], 24, 24);
            randstr (str24[8], 24, 24);
            randstr (str24[9], 24, 24);
            randdatastr (s_data[i], 26, 50);

            if (gen) {
               printf ("%d %d %d %s %s %s %s %s %s %s %s %s 0 0 0 %s\n",
                       sid, swid, s_quantity[i], str24[0], str24[1], str24[2],
                       str24[3], str24[4], str24[5], str24[6], str24[7],
                       str24[8], str24[9], s_data[i]);
            }
            else {
               s_i_id[i] = sid;
               s_w_id[i] = swid;
               strncpy (s_dist_01[i], str24[0], 24);
               strncpy (s_dist_02[i], str24[1], 24);
               strncpy (s_dist_03[i], str24[2], 24);
               strncpy (s_dist_04[i], str24[3], 24);
               strncpy (s_dist_05[i], str24[4], 24);
               strncpy (s_dist_06[i], str24[5], 24);
               strncpy (s_dist_07[i], str24[6], 24);
               strncpy (s_dist_08[i], str24[7], 24);
               strncpy (s_dist_09[i], str24[8], 24);
               strncpy (s_dist_10[i], str24[9], 24);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curs, STOCARR, 0)) {
               errrpt (&tpclda, &curs);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n", s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n", s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-----------------------------------------------------------------+
 | Load the STOCK table (cluster around s_i_id).                   |
 +-----------------------------------------------------------------*/

   if (do_S) {
```

```
      nrows = (eitem - bitem + 1) * (eware - bware + 1);

      fprintf (stderr, "Loading/generating stock: i%d - i%d, w%d - w%d (%d rows)\n
",
               bitem, eitem, bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      sid = bitem;
      swid = bware - 1;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < STOCARR; i++, row++) {
            if (++swid > eware) {          /* cheap mod */
               swid = bware;
               sid++;
            }
            s_quantity[i] = (rand () % 91) + 10;
            randstr (str24[0], 24, 24);
            randstr (str24[1], 24, 24);
            randstr (str24[2], 24, 24);
            randstr (str24[3], 24, 24);
            randstr (str24[4], 24, 24);
            randstr (str24[5], 24, 24);
            randstr (str24[6], 24, 24);
            randstr (str24[7], 24, 24);
            randstr (str24[8], 24, 24);
            randstr (str24[9], 24, 24);
            randdatastr (s_data[i], 26, 50);

            if (gen) {
               printf ("%d %d %d %s %s %s %s %s %s %s %s %s %s 0 0 0 %s\n",
                       sid, swid, s_quantity[i], str24[0], str24[1], str24[2],
                       str24[3], str24[4], str24[5], str24[6], str24[7],
                       str24[8], str24[9], s_data[i]);
            }
            else {
               s_i_id[i] = sid;
               s_w_id[i] = swid;
               strncpy (s_dist_01[i], str24[0], 24);
               strncpy (s_dist_02[i], str24[1], 24);
               strncpy (s_dist_03[i], str24[2], 24);
               strncpy (s_dist_04[i], str24[3], 24);
               strncpy (s_dist_05[i], str24[4], 24);
               strncpy (s_dist_06[i], str24[5], 24);
               strncpy (s_dist_07[i], str24[6], 24);
               strncpy (s_dist_08[i], str24[7], 24);
               strncpy (s_dist_09[i], str24[8], 24);
               strncpy (s_dist_10[i], str24[9], 24);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curs, STOCARR, 0)) {
               errrpt (&tpclda, &curs);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n", s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n", s_w_id[0],
                        s_i_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*---------------------------------------------------------------+
 | Load the HISTORY table.                                       |
 +---------------------------------------------------------------*/

   if (do_A || do_h) {
      nrows = (eware - bware + 1) * HISTFAC;

      fprintf (stderr, "Loading/generating history: w%d - w%d (%d rows)\n   ",
               bware, eware, nrows);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < HISTARR; i++, row++) {
            cid++;
            if (cid > CUSTFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift district cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift warehouse cycle */
               }
            }
            h_c_id[i] = cid;
            h_d_id[i] = cdid;
            h_w_id[i] = cwid;
            randstr (h_data[i], 12, 24);
            if (gen) {
               printf ("%d %d %d %d %d %s 1000 %s\n", cid, cdid, cwid, cdid,
                       cwid, sdate, h_data[i]);
            }
         }

         if (gen) {
            fflush (stdout);
         }
         else {
            if (oexn (&curh, HISTARR, 0)) {
               errrpt (&tpclda, &curh);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        h_w_id[0], h_d_id[0], h_c_id[0]);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                        h_w_id[0], h_d_id[0], h_c_id[0]);
               quit ();
               exit (1);
            }
         }

         if ((++loopcount) % 50)
            fprintf (stderr, ".");
         else
            fprintf (stderr, " %d rows committed\n   ", row);
      }

      end_time = gettime ();
```

```
      end_cpu = getcpu ();
      fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
               nrows, end_time - begin_time, end_cpu - begin_cpu);
   }

/*-------------------------------------------------------------+
 | Load the ORDERS and ORDER-LINE table.                       |
 +-------------------------------------------------------------*/

   if (do_A || do_o) {
      nrows = (eware - bware + 1) * ORDEFAC * DISTFAC;

      fprintf (stderr, "Loading/generating orders and order-line: w%d - w%d (%d ord,
~%d ordl)\n   ",
               bware, eware, nrows, nrows * 10);

      begin_time = gettime ();
      begin_cpu = getcpu ();

      cid = 0;
      cdid = 1;
      cwid = bware;
      loopcount = 0;

      for (row = 0; row < nrows; ) {
         for (i = 0; i < ORDEARR; i++, row++) {
            cid++;
            if (cid > ORDEFAC) {        /* cycle cust id */
               cid = 1;                 /* cheap mod */
               cdid++;                  /* shift district cycle */
               if (cdid > DISTFAC) {
                  cdid = 1;
                  cwid++;               /* shift warehouse cycle */
               }
            }
            o_carrier_id[i] = rand () % 10 + 1;
            o_ol_cnt[i] = olcnt = rand () % 11 + 5;

            if (gen) {
               if (cid < 2101) {
                  printf ("%d %d %d %d %s %d %d 1\n", cid, cdid, cwid,
                          randperm3000[cid - 1], sdate,o_carrier_id[i],
                          o_ol_cnt[i]);
               }
               else {
                   /* set carrierid to 11 instead of null */
                  printf ("%d %d %d %d %s 11 %d 1\n", cid, cdid, cwid,
                          randperm3000[cid - 1], sdate, o_ol_cnt[i]);
               }
            }
            else {
               o_id[i] = cid;
               o_d_id[i] = cdid;
               o_w_id[i] = cwid;
               o_c_id[i] = randperm3000[cid - 1];
            }

            for (j = 0; j < o_ol_cnt[i]; j++) {
               ol_i_id[j] = sid = lrand48 () % 100000 + 1;
               if (cid < 2101)
                  ol_amount[j] = 0;
               else
                  ol_amount[j] = (lrand48 () % 999999 + 1) ;
               randstr (str24[j], 24, 24);

               if (gen) {
                  if (cid < 2101) {
                     fprintf (olfp, "%d %d %d %d %s %d %d 5 %ld %s\n", cid,
                              cdid, cwid, j + 1, sdate, ol_i_id[j], cwid,
                              ol_amount[j], str24[j]);
                  }
                  else {
                       /* Insert a default date instead of null date */
                     fprintf (olfp, "%d %d %d %d 01-Jan-1811 %d %d 5 %ld %s\n", cid,
                              cdid, cwid, j + 1, ol_i_id[j], cwid,
                              ol_amount[j], str24[j]);
                  }
               }
               else {
                  ol_o_id[j] = cid;
                  ol_d_id[j] = cdid;
                  ol_w_id[j] = cwid;
                  ol_number[j] = j + 1;
                  ol_supply_w_id[j] = cwid;
                  strncpy (ol_dist_info[j], str24[j], 24);
               }
            }

            if (gen) {
               fflush (olfp);
            }
            else {
               if (cid < 2101) {
                  if (oexn (&curol1, olcnt, 0)) {
                     errrpt (&tpclda, &curol1);
                     orol (&tpclda);
                     fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n",
                              cwid, cdid, cid);
                     quit ();
                     exit (1);
                  }
                  else if (ocom (&tpclda)) {
                     errrpt (&tpclda, &tpclda);
                     orol (&tpclda);
                     fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n",
                              cwid, cdid, cid);
                     quit ();
                     exit (1);
                  }
               }
               else {
                  if (oexn (&curol2, olcnt, 0)) {
                     errrpt (&tpclda, &curol2);
                     orol (&tpclda);
                     fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n",
                              cwid, cdid, cid);
                     quit ();
                     exit (1);
                  }
                  else if (ocom (&tpclda)) {
                     errrpt (&tpclda, &tpclda);
                     orol (&tpclda);
                     fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n",
                              cwid, cdid, cid);
                     quit ();
                     exit (1);
                  }
               }
            }
         }
      }

      if (gen) {
         fflush (stdout);
      }
      else {
         if (cid < 2101) {
            if (oexn (&curo1, ORDEARR, 0)) {
               errrpt (&tpclda, &curo1);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n   ",
                        cwid, cdid, cid);
               quit ();
               exit (1);
            }
            else if (ocom (&tpclda)) {
               errrpt (&tpclda, &tpclda);
               orol (&tpclda);
               fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n   ",
                        cwid, cdid, cid);
               quit ();
               exit (1);
            }
         }
         else {
            if (oexn (&curo2, ORDEARR, 0)) {
               errrpt (&tpclda, &curo2);
               orol (&tpclda);
```

```
                fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n   ",
                         cwid, cdid, cid);
                quit ();
                exit (1);
             }
             else if (ocom (&tpclda)) {
                errrpt (&tpclda, &tpclda);
                orol (&tpclda);
                fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n   ",
                         cwid, cdid, cid);
                quit ();
                exit (1);
             }
          }
       }

       if ((++loopcount) % 50)
          fprintf (stderr, ".");
       else
          fprintf (stderr, " %d orders committed\n   ", row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d orders loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
             nrows, end_time - begin_time, end_cpu - begin_cpu);
 }

/*-------------------------------------------------------------+
 | Load the NEW-ORDER table.                                   |
 +-------------------------------------------------------------*/

 if (do_A || do_n) {
    nrows = (eware - bware + 1) * NEWOFAC * DISTFAC;

    fprintf (stderr, "Loading/generating new-order: w%d - w%d (%d rows)\n   ",
             bware, eware, nrows);

    begin_time = gettime ();
    begin_cpu = getcpu ();

    cid = 0;
    cdid = 1;
    cwid = bware;
    loopcount = 0;

    for (row = 0; row < nrows; ) {
       for (i = 0; i < NEWOARR; i++, row++) {
          cid++;
          if (cid > NEWOFAC) {
             cid = 1;
             cdid++;
             if (cdid > DISTFAC) {
                cdid = 1;
                cwid++;
             }
          }

          if (gen) {
             printf ("%d %d %d\n", cid + 2100, cdid, cwid);
          }
          else {
             no_o_id[i] = cid + 2100;
             no_d_id[i] = cdid;
             no_w_id[i] = cwid;
          }
       }

       if (gen) {
          fflush (stdout);
       }
       else {
          if (oexn (&curno, NEWOARR, 0)) {
             errrpt (&tpclda, &curno);
             orol (&tpclda);
             fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n   ",
                      cwid, cdid, cid + 2100);
             quit ();
             exit (1);
          }
          else if (ocom (&tpclda)) {
             errrpt (&tpclda, &tpclda);
             orol (&tpclda);
             fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n   ",
                      cwid, cdid, cid + 2100);
             quit ();
             exit (1);
          }
       }

       if ((++loopcount) % 45)
          fprintf (stderr, ".");
       else
          fprintf (stderr, " %d rows committed\n   ", row);
    }

    end_time = gettime ();
    end_cpu = getcpu ();
    fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f
cpu)\n\n",
             nrows, end_time - begin_time, end_cpu - begin_cpu);
 }

/*-------------------------------------------------------------+
 | clean up and exit.                                          |
 +-------------------------------------------------------------*/

 if (olfp)
    fclose (olfp);
 if (!gen)
    quit ();
 exit (0);

}


initperm ()

{

 int i;
 int pos;
 int temp;

 /* init randperm3000 */

 for (i = 0; i < 3000; i++)
    randperm3000[i] = i + 1;
 for (i = 3000; i > 0; i--) {
    pos = rand () % i;
    temp = randperm3000[i - 1];
    randperm3000[i - 1] = randperm3000[pos];
    randperm3000[pos] = temp;
 }

}


randstr (str, x, y)

char *str;
int x;
int y;

{

 int i, j;
 int len;

 len = (rand () % (y - x + 1)) + x;
 for (i = 0; i < len; i++) {
    j = rand () % 62;
    if (j < 26)
       str[i] = (char) (j + 'a');
    else if (j < 52)
```

```
         str[i] = (char) (j - 26 + 'A');
      else
         str[i] = (char) (j - 52 + '0');
   }
   str[len] = '\0';

}


randdatastr (str, x, y)

char *str;
int x;
int y;

{

   int i, j;
   int len;
   int pos;

   len = (rand () % (y - x + 1)) + x;
   for (i = 0; i < len; i++) {
      j = rand () % 62;
      if (j < 26)
         str[i] = (char) (j + 'a');
      else if (j < 52)
         str[i] = (char) (j - 26 + 'A');
      else
         str[i] = (char) (j - 52 + '0');
   }
   str[len] = '\0';
   if ((rand () % 10) == 0) {
      pos = (rand () % (len - 8));
      str[pos] = 'O';
      str[pos + 1] = 'R';
      str[pos + 2] = 'I';
      str[pos + 3] = 'G';
      str[pos + 4] = 'I';
      str[pos + 5] = 'N';
      str[pos + 6] = 'A';
      str[pos + 7] = 'L';
   }

}



randnum (str, len)

char *str;
int len;

{

   int i;

   for (i = 0; i < len; i++)
      str[i] = (char) (rand () % 10 + '0');
   str[len] = '\0';

}



randlastname (str, id)

char *str;
int id;

{

   id = id % 1000;
   strcpy (str, lastname[id / 100]);
   strcat (str, lastname[(id / 10) % 10]);
   strcat (str, lastname[id % 10]);

}


NURand (A, x, y, cnum)

int A, x, y, cnum;

{

   int a, b;

   a = lrand48 () % (A + 1);
   b = (lrand48 () % (y - x + 1)) + x;
   return ((((a | b) + cnum) % (y - x + 1)) + x);

}



sysdate (sdate)

char *sdate;

{

   time_t tp;
   struct tm *tmptr;

   time (&tp);
   tmptr = localtime (&tp);
   strftime (sdate, 29, "%d-%b-%Y", tmptr);

}
```

build/loader/dist.ctl

```
--
-- $Header: cust.ctl 7030100.1 95/08/07 15:46:36 plai Generic<base> $ Copyr (c) 1994
Oracle
--
-- +-------------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA         |
--                      OPEN SYSTEMS PERFORMANCE GROUP                 |
--                           All Rights Reserved                       |
-- +-------------------------------------------------------------------+
--  FILENAME
--      cust.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading customers to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- ===================================================================*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE district
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(

 D_ID                                    integer external,
 D_W_ID                                  integer external,
 D_YTD                                   integer external,
 D_TAX                                   float external,
 D_NEXT_O_ID                             integer external,
 D_NAME                                  CHAR(10),
 D_STREET_1                              CHAR(20),
 D_STREET_2                              CHAR(20),
 D_CITY                                  CHAR(20),
 D_STATE                                 CHAR(2),
 D_ZIP                                   CHAR(9)
```

```
)
```

build/loader/hist.ctl

```
--
-- $Header: hist.ctl 7030100.1 95/08/07 15:47:23 plai Generic<base> $ Copyr (c) 1994
Oracle
--
-- +--------------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA          |
--                      OPEN SYSTEMS PERFORMANCE GROUP                  |
--                           All Rights Reserved                        |
-- +--------------------------------------------------------------------+
--  FILENAME
--      hist.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading history rows to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- ====================================================================*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE history
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   h_c_id              integer external,
   h_c_d_id            integer external,
   h_c_w_id            integer external,
   h_d_id              integer external,
   h_w_id              integer external,
   h_date              date "DD-Mon-YYYY",
   h_amount            integer external,
   h_data              char(24)
)
```

build/loader/neword.ctl

```
--
-- $Header: neword.ctl 7030100.1 95/08/07 15:47:44 plai Generic<base> $ Copyr (c)
1994 Oracle
--
-- +--------------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA          |
--                      OPEN SYSTEMS PERFORMANCE GROUP                  |
--                           All Rights Reserved                        |
-- +--------------------------------------------------------------------+
--  FILENAME
--      neword.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading new orders to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- ====================================================================*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE new_order
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   no_o_id             integer external,
   no_d_id             integer external,
   no_w_id             integer external
)
```

build/loader/order.ctl

```
--
-- $Header: order.ctl 7030100.1 95/08/07 15:54:01 plai Generic<base> $ Copyr (c)
1994 Oracle
--
-- +--------------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA          |
--                      OPEN SYSTEMS PERFORMANCE GROUP                  |
--                           All Rights Reserved                        |
-- +--------------------------------------------------------------------+
--  FILENAME
--      order.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading orders to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- ====================================================================*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE orders
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   o_id                integer external,
   o_d_id              integer external,
   o_w_id              integer external,
   o_c_id              integer external,
   o_entry_d           date "DD-Mon-YYYY",
   o_carrier_id        integer external,
   o_ol_cnt            integer external,
   o_all_local         integer external
)
```

build/loader/ordline.ctl

```
   --
-- $Header: ordline.ctl 7030100.1 95/08/07 15:54:11 plai Generic<base> $ Copyr (c)
1994 Oracle
--
-- +--------------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA          |
--                      OPEN SYSTEMS PERFORMANCE GROUP                  |
--                           All Rights Reserved                        |
-- +--------------------------------------------------------------------+
--  FILENAME
--      ordline.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading order lines to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- ====================================================================*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE
```

```
LOAD DATA
APPEND

INTO TABLE order_line
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   ol_o_id             integer external,
   ol_d_id             integer external,
   ol_w_id             integer external,
   ol_number           integer external,
   ol_delivery_d       date "DD-Mon-YYYY",
   ol_i_id             integer external,
   ol_supply_w_id      integer external,
   ol_quantity         integer external,
   ol_amount           integer external,
   ol_dist_info        char(24)
)
```

build/loader/stock.ctl

```
--
-- $Header: stock.ctl 7030100.1 95/08/07 15:54:18 plai Osd<base> $ Copyr (c) 1994
Oracle
--
-- +-------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA        |
--                      OPEN SYSTEMS PERFORMANCE GROUP                |
--                           All Rights Reserved                      |
-- +-------------------------------------------------------------+
--  FILENAME
--      stock.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading stocks to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- +-------------------------------------------------------------+*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE stock
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   s_i_id              integer external,
   s_w_id              integer external,
   s_quantity          integer external,
   s_dist_01           char(24),
   s_dist_02           char(24),
   s_dist_03           char(24),
   s_dist_04           char(24),
   s_dist_05           char(24),
   s_dist_06           char(24),
   s_dist_07           char(24),
   s_dist_08           char(24),
   s_dist_09           char(24),
   s_dist_10           char(24),
   s_ytd               integer external,
   s_order_cnt         integer external,
   s_remote_cnt        integer external,
   s_data              char(50)
)
```

build/loader/ware.ctl

```
--
-- $Header: cust.ctl 7030100.1 95/08/07 15:46:36 plai Generic<base> $ Copyr (c) 1994
Oracle
--
-- +-------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA        |
--                      OPEN SYSTEMS PERFORMANCE GROUP                |
--                           All Rights Reserved                      |
-- +-------------------------------------------------------------+
--  FILENAME
--      cust.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading customers to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- +-------------------------------------------------------------+*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE warehouse
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(

 W_ID                                    integer external,
 W_YTD                                   integer external,
 W_TAX                                   float external,
 W_NAME                                  CHAR(10),
 W_STREET_1                              CHAR(20),
 W_STREET_2                              CHAR(20),
 W_CITY                                  CHAR(20),
 W_STATE                                 CHAR(2),
 W_ZIP                                   CHAR(9)
)
```

build/loader/item.ctl

```
-- +-------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA        |
--                      OPEN SYSTEMS PERFORMANCE GROUP                |
--                           All Rights Reserved                      |
-- +-------------------------------------------------------------+
--  FILENAME
--      cust.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading customers to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- +-------------------------------------------------------------+*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE item
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(

 I_ID                                    integer external,
 I_IM_ID                                 integer external,
 I_NAME                                  CHAR(24),
 I_PRICE                                 integer external,
```

```
 I_DATA                                  CHAR(50)
)
```

build/loader/cust.ctl

```
--
-- $Header: cust.ctl 7030100.1 95/08/07 15:46:36 plai Generic<base> $ Copyr (c) 1994
Oracle
--
-- +-------------------------------------------------------------------+
--         Copyright (c) 1994  Oracle Corp, Redwood Shores, CA         |
--                      OPEN SYSTEMS PERFORMANCE GROUP                 |
--                           All Rights Reserved                       |
-- +-------------------------------------------------------------------+
--  FILENAME
--      cust.ctl
--  DESCRIPTION
--      This is a SQL*Loader control file.  It is used for
--      loading customers to the tpcc database.
--  USAGE
--      sqlldr[st] <user_name>/<password> <SQL*Loader control file>
-- +-------------------------------------------------------------------+*/

-- OPTIONS (DIRECT = TRUE, PARALLEL = TRUE)

-- UNRECOVERABLE

LOAD DATA
APPEND

INTO TABLE customer
APPEND
FIELDS TERMINATED BY WHITESPACE OPTIONALLY ENCLOSED BY '"'
(
   c_id                integer external,
   c_d_id              integer external,
   c_w_id              integer external,
   c_first             char(16),
   c_middle            char(2),
   c_last              char(16),
   c_street_1          char(20),
   c_street_2          char(20),
   c_city              char(20),
   c_state             char(2),
   c_zip               char(9),
   c_phone             char(16),
   c_since             date "DD-Mon-YYYY",
   c_credit            char(2),
   c_credit_lim        integer external,
   c_discount          integer external,
   c_balance           integer external,
   c_ytd_payment       integer external,
   c_payment_cnt       integer external,
   c_delivery_cnt      integer external,
   c_data              char(500)
)
```

10.3 Packages and Views

build/blocks/views.sql

```
create or replace view wh_cust
(w_id, w_tax, c_id, c_d_id, c_w_id, c_discount, c_last, c_credit)
as select w.w_id, w.w_tax,
          c.c_id, c.c_d_id, c.c_w_id, c.c_discount, c.c_last, c.c_credit
   from customer c, warehouse w
   where w.w_id = c.c_w_id
/

create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax )
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id, w.w_tax
   from district d, warehouse w
   where w.w_id = d.d_w_id
/

create  or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data, s_quantity,
 s_order_cnt, s_ytd, s_remote_cnt,
 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)
as
 select i.i_id, s_w_id, i.i_price, i.i_name, i.i_data, s_data, s_quantity,
 s_order_cnt, s_ytd, s_remote_cnt,
 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
  from stock s, item i
 where i.i_id = s.s_i_id
/
exit
```

build/blocks/initpay.sql

```
CREATE OR REPLACE PACKAGE initpay
AS
 TYPE rowidarray IS TABLE OF ROWID INDEX BY BINARY_INTEGER;
 row_id                  rowidarray;
 cust_rowid              ROWID;
 dist_name               VARCHAR2(11);
 ware_name               VARCHAR2(11);
 c_num                   BINARY_INTEGER;
 PROCEDURE pay_init;
END initpay;
/
CREATE OR REPLACE PACKAGE BODY initpay AS
  PROCEDURE pay_init IS
  BEGIN
    NULL;
  END pay_init;
END initpay;
/
exit
```

build/blocks/tkvcinin.sql

```
CREATE OR REPLACE PACKAGE initnew
AS
 TYPE intarray IS TABLE OF INTEGER index by binary_integer;
 TYPE distarray IS TABLE OF VARCHAR(24) index by binary_integer;
 nulldate       DATE;
 s_dist                        distarray;
 idx1arr            intarray;
 s_remote           intarray;
 PROCEDURE new_init(idxarr intarray);
END initnew;
/
show errors;

CREATE OR REPLACE PACKAGE BODY initnew AS
  PROCEDURE new_init (idxarr  intarray)
  IS
  BEGIN
   nulldate := TO_DATE('01-01-1811', 'MM-DD-YYYY');
    idx1arr := idxarr;
  END new_init;
END initnew;
/
show errors
exit
```

Appendix C Tunable Parameters

The HP-UX operating system tunable parameters employed to generate the kernel for the HP 9000 V2500 Enterprise Server and the 24 HP 9000 Model C360 clients are listed below. Included as well are the Oracle8i Enterprise Database Server v8.1.5 and TUXEDO 6.4 paramters.

C.1 HP-UX Configuration - Clients

Config/Client4/ostune.ver

```
**************************************
* $Source: /usr/local/kcs/sys.ROSE_800/filesets.info/CORE-KRN/RCS/generic,v $
* $Revision: 1.3.106.2 $      $Author: kcs $
* $State: Exp $     $Locker: CRT $
* $Date: 97/07/12 21:51:58 $
*
**************************************
* Additional drivers required in every machine-type to create a complete
* system file during cold install.  This list is every driver that the
* master.d/ files do not force on the system or is not identifiable by
* ioscan.
* Other CPU-type specific files can exist for their special cases.
* see create_sysfile (1m).
**************************************
*
* Drivers/Subsystems
ccio
GSCtoPCI
c720
sdisk
sctl
asio0
cdfs
dlpi
inet
uipc
tun
telm
tels
netdiag1
nms
lvm
lv
hpstreams
clone
strlog
sad
echo
sc
timod
tirdwr
pipedev
pipemod
ffs
ldterm
ptem
pts
ptm
pckt
diag1
token_arp
btlan3
maclan
diag2
dmem
dev_config
btlan5


* Kernel Device info

dump lvol


* Tunable parameters

default_disk_ir 1
STRMSGSZ        65535
bufpages        1024
fs_async        1
maxfiles        2048
maxfiles_lim    2048
maxssiz         0X8000000
maxswapchunks   2048
maxuprc         3600
maxusers        3500
msgmap          (MSGTQL+2)
msgmax          32768
msgmnb          65535
msgmni          (NPROC)
msgseg          (MSGTQL*4)
msgssz          512
msgtql          (NPROC)
*nfile           16000
nfile           20000
nflocks         4096
ninode          6000
nproc           3700
npty            100
nstrpty         60
semmni          (NPROC)
semmns          (SEMMNI)
semmnu          (SEMMNI)
semvmx          32768
shmmax          1073741824
shmmni          16
shmseg          16
swapmem_on      0
timezone        480
unlockable_mem  1
```

C.2 HP-UX Configurtion – Server

Config/Server/ostune.ver

```
**************************************
* $Source: /usr/local/kcs/sys.ROSE_800/filesets.info/CORE-KRN/RCS/generic,v $
* $Revision: 1.3.106.2 $        $Author: kcs $
* $State: Exp $         $Locker: CRT $
* $Date: 97/07/12 21:51:58 $
*
**************************************
* Additional drivers required in every machine-type to create a complete
* system file during cold install.  This list is every driver that the
* master.d/ files do not force on the system or is not identifiable by
* ioscan.
* Other CPU-type specific files can exist for their special cases.
* see create_sysfile (1m).
**************************************
*
* Drivers/Subsystems
epic
c720
```

```
sdisk
sctl
stape
conspl
cdfs
dlpi
inet
uipc
tun
telm
tels
netdiag1
nms
btlan6
fcT1_fcp
fcT1_cntl
fcpmux
fcT1
fcp
fcms
gelan
lvm
lv
hpstreams
clone
strlog
sad
echo
sc
timod
tirdwr
pipedev
pipemod
ffs
ldterm
ptem
pts
ptm
pckt
token_arp
diag0
diag2
dmem
dev_config
prf
onyxe
coke
asyncdsk
dump lvol
**********
* Tunables
*********
STRMSGSZ            65535
nstrpty             60
maxfiles            2048
maxfiles_lim        2048
nflocks             2048
fs_async            0
bufpages            1024
eqmemsize           512
maxusers            1024
maxuprc             900
max_async_ports     1024
nproc               1024
nhtbl_scale         1
maxswapchunks       16384
nfile               170000
ninode              15000
npty                10
shmmni              104
semmni              2048
semmns              2048
semmnu              2048
shmmax              34359738368
shmseg              16
maxssiz             0x10000000
maxdsiz             0x30000000
timezone            480
maxvgs              64
num_tachyon_adapters            12
max_fcp_reqs        1024
unlockable_mem      1
swapmem_on          0
```

C.3 Oracle8i Enterprise Database Server v8.1.5 Parameters

Config/Server/dbtune.ver

```
#
# $Header: p_run.ora 7030100.2 96/05/02 19:22:28 plai Generic<base> $ Copyr (c) 1993
Oracle
#
#===================================================================+
#        Copyright (c) 1996  Oracle Corp, Redwood Shores, CA        |
#                    OPEN SYSTEMS PERFORMANCE GROUP                 |
#                       All Rights Reserved                         |
#===================================================================+
# FILENAME
#      p_run.ora
# DESCRIPTION
#     Oracle parameter file for running TPC-C.
#====================================================================

control_files                           = (?/dbs/tpcc_disks/control01,
?/dbs/tpcc_disks/control02)
db_writer_processes             = 4
cpu_count                                 = 32
disk_asynch_io                  = TRUE
lock_sga                        = TRUE
db_block_lru_latches            = 32
_spin_count                      = 40000
parallel_max_servers            = 220
recovery_parallelism            = 300
compatible                      = 8.1.4.0.0
db_name                         = tpcc
db_files                        = 300
db_file_multiblock_read_count   = 32
db_block_buffers                = 13000000
_db_block_hash_buckets          = 26000000
buffer_pool_recycle  = ( buffers:120000, lru_latches:12 )
buffer_pool_keep     = ( buffers:10000000, lru_latches:8 )
db_block_max_dirty_target       = 8000000
fast_start_io_target            = 0
_disable_incremental_checkpoints  = TRUE
_db_aging_hot_criteria          = 2
_db_aging_stay_count            = 1
_db_writer_chunk_writes                 = 1000
_db_writer_max_writes                   = 1000
dml_locks                       = 500
enqueue_resources               = 600000
hash_join_enabled               = FALSE
log_archive_start               = FALSE
_log_archive_buffer_size        = 32
log_checkpoint_timeout          = 0
log_checkpoint_interval         = 1000000000
log_checkpoints_to_alert        = TRUE
log_buffer                      = 33554432
_log_simultaneous_copies        = 256
open_cursors                    = 700
processes                       = 750
sessions                        = 750
transactions                    = 1000
distributed_transactions        = 0
transactions_per_rollback_segment = 1
max_rollback_segments           = 900
rollback_segments               = (t1,t2,t3,t4,t5,t6,t7,t8,t9,
t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,
```

```
t20,t21,t22,t23,t24,t25,t26,t27,t28,t29,
t30,t31,t32,t33,t34,t35,t36,t37,t38,t39,
t40,t41,t42,t43,t44,t45,t46,t47,t48,t49,
t50,t51,t52,t53,t54,t55,t56,t57,t58,t59,
t60,t61,t62,t63,t64,t65,t66,t67,t68,t69,
t70,t71,t72,t73,t74,t75,t76,t77,t78,t79,
t80,t81,t82,t83,t84,t85,t86,t87,t88,t89,
t90,t91,t92,t93,t94,t95,t96,t97,t98,t99,
t100,t101,t102,t103,t104,t105,t106,t107,t108,t109,
t110,t111,t112,t113,t114,t115,t116,t117,t118,t119,
t120,t121,t122,t123,t124,t125,t126,t127,t128,t129,
t130,t131,t132,t133,t134,t135,t136,t137,t138,t139,
t140,t141,t142,t143,t144,t145,t146,t147,t148,t149,
t150,t151,t152,t153,t154,t155,t156,t157,t158,t159,
t160,t161,t162,t163,t164,t165,t166,t167,t168,t169,
t170,t171,t172,t173,t174,t175,t176,t177,t178,t179,
t180,t181,t182,t183,t184,t185,t186,t187,t188,t189,
t190,t191,t192,t193,t194,t195,t196,t197,t198,t199,
t200)
rollback_segments          = (t201,t202,t203,t204,t205,t206,t207,t208,t209,
t210,t211,t212,t213,t214,t215,t216,t217,t218,t219,
t220,t221,t222,t223,t224,t225,t226,t227,t228,t229,
t230,t231,t232,t233,t234,t235,t236,t237,t238,t239,
t240,t241,t242,t243,t244,t245,t246,t247,t248,t249,
t250,t251,t252,t253,t254,t255,t256,t257,t258,t259,
t260,t261,t262,t263,t264,t265,t266,t267,t268,t269,
t270,t271,t272,t273,t274,t275,t276,t277,t278,t279,
t280,t281,t282,t283,t284,t285,t286,t287,t288,t289,
t290,t291,t292,t293,t294,t295,t296,t297,t298,t299,
t300,t301,t302,t303,t304,t305,t306,t307,t308,t309,
t310,t311,t312,t313,t314,t315,t316,t317,t318,t319,
t320,t321,t322,t323,t324,t325,t326,t327,t328,t329,
t330,t331,t332,t333,t334,t335,t336,t337,t338,t339,
t340,t341,t342,t343,t344,t345,t346,t347,t348,t349,
t350,t351,t352,t353,t354,t355,t356,t357,t358,t359,
t360,t361,t362,t363,t364,t365,t366,t367,t368,t369,
t370,t371,t372,t373,t374,t375,t376,t377,t378,t379,
t380,t381,t382,t383,t384,t385,t386,t387,t388,t389,
t390,t391,t392,t393,t394,t395,t396,t397,t398,t399,
t400)
rollback_segments          = (t401,t402,t403,t404,t405,t406,t407,t408,t409,
t410,t411,t412,t413,t414,t415,t416,t417,t418,t419,
t420,t421,t422,t423,t424,t425,t426,t427,t428,t429,
t430,t431,t432,t433,t434,t435,t436,t437,t438,t439,
t440,t441,t442,t443,t444,t445,t446,t447,t448,t449,
t450,t451,t452,t453,t454,t455,t456,t457,t458,t459,
t460,t461,t462,t463,t464,t465,t466,t467,t468,t469,
t470,t471,t472,t473,t474,t475,t476,t477,t478,t479,
t480,t481,t482,t483,t484,t485,t486,t487,t488,t489,
t490,t491,t492,t493,t494,t495,t496,t497,t498,t499,
t500,t501,t502,t503,t504,t505,t506,t507,t508,t509,
t510,t511,t512,t513,t514,t515,t516,t517,t518,t519,
t520,t521,t522,t523,t524,t525,t526,t527,t528,t529,
t530,t531,t532,t533,t534,t535,t536,t537,t538,t539,
t540,t541,t542,t543,t544,t545,t546,t547,t548,t549,
t550,t551,t552,t553,t554,t555,t556,t557,t558,t559,
t560,t561,t562,t563,t564,t565,t566,t567,t568,t569,
t570,t571,t572,t573,t574,t575,t576,t577,t578,t579,
t580,t581,t582,t583,t584,t585,t586,t587,t588,t589,
t590,t591,t592,t593,t594,t595,t596,t597,t598,t599,
t600)
rollback_segments          = (t601,t602,t603,t604,t605,t606,t607,t608,t609,
t610,t611,t612,t613,t614,t615,t616,t617,t618,t619,
t620,t621,t622,t623,t624,t625,t626,t627,t628,t629,
t630,t631,t632,t633,t634,t635,t636,t637,t638,t639,
t640,t641,t642,t643,t644,t645,t646,t647,t648,t649,
t650,t651,t652,t653,t654,t655,t656,t657,t658,t659,
t660,t661,t662,t663,t664,t665,t666,t667,t668,t669,
t670,t671,t672,t673,t674,t675,t676,t677,t678,t679,
t680,t681,t682,t683,t684,t685,t686,t687,t688,t689,
t690,t691,t692,t693,t694,t695,t696,t697,t698,t699,
t700,t701,t702,t703,t704,t705,t706,t707,t708,t709,
t710,t711,t712,t713,t714,t715,t716,t717,t718,t719,
t720,t721,t722,t723,t724,t725,t726,t727,t728,t729,
t730,t731,t732,t733,t734,t735,t736,t737,t738,t739,
t740,t741,t742,t743,t744,t745,t746,t747,t748,t749,
t750,t751,t752,t753,t754,t755,t756,t757,t758,t759,
t760,t761,t762,t763,t764,t765,t766,t767,t768,t769,
t770,t771,t772,t773,t774,t775,t776,t777,t778,t779,
t780,t781,t782,t783,t784,t785,t786,t787,t788,t789,
t790,t791,t792,t793,t794,t795,t796,t797,t798,t799,
t800)
shared_pool_size                = 100000000
shared_pool_reserved_size       = 20000000
_discrete_transactions_enabled   = FALSE
cursor_space_for_time           = TRUE
sort_area_size                            = 524288
gc_releasable_locks           = 0
transaction_auditing            = FALSE
replication_dependency_tracking   = FALSE
_db_block_cache_protect         = FALSE
db_block_checking               = FALSE
max_dump_file_size              = 10K
```

C.4 Tuxedo UBBconfig

Config/Client4/tmcfg.ver

```
# This is a UBBconfig for a client1-server configuration.
#
# This UBBconfig requires settings for:
# SERVER_NAME CLIENT_NAME MASTER_NAME SERVER_ADDR CLIENT_ADDR NODE_NAMES
# TLISTEN_PORT TBRIDGE_PORT
# In addition, it requires setting the things all UBBconfig.gens need:
#         IPCKEY              some decent IPCKEY, should be different for each
config
#         ROOTDIR
#         TUXCONFIG
#         APPDIR
#         ULOGDIR
#

#--------------------------------------------------------------------
*RESOURCES
#--------------------------------------------------------------------
          IPCKEY    40001
          PERM      0666
          MASTER    client4

   MAXACCESSERS     3450      # 1024 or more
   MAXGTT           1024
   MAXSERVERS       28
   MAXSERVICES      125   # MAXSERVERS * #-of-services-each-server + 10( for BBL)
   MODEL  SHM
   LDBAL  Y

# During benchmark, don't want to scan too often.  In particular, while
# the client1s are stabilizing in virtual memory, we don't want to sanity
# scan; and if we do sanity scan, we want large timeouts, since the BRIDGE
# the BBL, the DBBL, and the client1s aren't getting much CPU time during that
# period.  Current settings:
#         *  scan servers every 5 minutes (maximum allowed by TUXEDO);
#         *  wait 1 minute for sanity  responses (maximum allowed by TUXEDO);
#         *  scan all the BBLs from DBBL every 30 minutes (want one scan in the
#            audited results);
#         *  timeout a blocking call after 5 minutes (the maximum).
          SCANUNIT  60
          SANITYSCAN          5
          DBBLWAIT  1
          BBLQUERY  30
          BLOCKTIME 5
#
#--------------------------------------------------------------------
*MACHINES
#--------------------------------------------------------------------
DEFAULT:
               TUXCONFIG="/project/iti/confs/TUXconfig.client4"
                   ROOTDIR="/project/iti"
```

```
                  APPDIR="/project/tpcc/bin"
                  ULOGPFX="/tmp/TUXEDO_LOG"

# for debugging, put both into the same log on the same machine
#                 ULOGPFX="/home/iti/confs/tpcc/ULOG"
# but for a big run, need some space, and want them local to the
# machine rather than across the net.

# Leave TUXCONFIG alone on the MASTER machine; over-ride for each
# other machine?
client4   LMID=client4
        TUXCONFIG="/project/iti/confs/TUXconfig.client4"
#-----------------------------------------------------------------
*GROUPS
#-----------------------------------------------------------------
group1            LMID=client4
                   GRPNO=1
group2            LMID=client4
                   GRPNO=2
group3            LMID=client4
                   GRPNO=3
group4            LMID=client4
                   GRPNO=4
group5            LMID=client4
                   GRPNO=5

#-----------------------------------------------------------------
#-----------------------------------------------------------------

#-----------------------------------------------------------------
*SERVERS
#-----------------------------------------------------------------
#
# "--" is application-specific arguments to be passed to server
# "-n" is designed to specify server-id

service SRVGRP=group1
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n1"
        RQADDR=tpcc_1 SRVID=1

service SRVGRP=group1
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n2"
        RQADDR=tpcc_2 SRVID=2

service SRVGRP=group1
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n3"
        RQADDR=tpcc_3 SRVID=3

service SRVGRP=group1
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n4"
        RQADDR=tpcc_4 SRVID=4

service SRVGRP=group1
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n5"
        RQADDR=tpcc_5 SRVID=5

service SRVGRP=group2
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n6"
        RQADDR=tpcc_6 SRVID=6

service SRVGRP=group2
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n7"
        RQADDR=tpcc_7 SRVID=7

service SRVGRP=group2
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n8"
        RQADDR=tpcc_8 SRVID=8

service SRVGRP=group2
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -n9"
        RQADDR=tpcc_9 SRVID=9

service SRVGRP=group2
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n10"
        RQADDR=tpcc_10 SRVID=10

service SRVGRP=group3
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n11"
        RQADDR=tpcc_11 SRVID=11

service SRVGRP=group3
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n12"
        RQADDR=tpcc_12 SRVID=12

service SRVGRP=group3
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n13"
        RQADDR=tpcc_13 SRVID=13

service SRVGRP=group3
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n14"
        RQADDR=tpcc_14 SRVID=14

service SRVGRP=group3
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n15"
        RQADDR=tpcc_15 SRVID=15

service SRVGRP=group4
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n16"
        RQADDR=tpcc_16 SRVID=16

service SRVGRP=group4
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n17"
        RQADDR=tpcc_17 SRVID=17

service SRVGRP=group4
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n18"
        RQADDR=tpcc_18 SRVID=18

service SRVGRP=group4
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n19"
        RQADDR=tpcc_19 SRVID=19

service SRVGRP=group4
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n20"
        RQADDR=tpcc_20 SRVID=20

service SRVGRP=group5
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n21"
        RQADDR=tpcc_21 SRVID=21

service SRVGRP=group5
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n22"
        RQADDR=tpcc_22 SRVID=22

service SRVGRP=group5
           CLOPT="-s NEWO_SVC -s PMT_SVC -s ORDS_SVC -s STKL_SVC -s DVRY_SVC -- -
n23"
        RQADDR=tpcc_23 SRVID=23
#-----------------------------------------------------------------
*SERVICES
#-----------------------------------------------------------------
*ROUTING
#-----------------------------------------------------------------
```

Appendix D RTE Configuration

This appendix lists RTE input parameters and code fragments used to generate each transaction input file, to demonstrate the RTE was configured to generate transaction input data as specified in *Clause 2* of the specification.

D.1 RTE Parameters

Run1/TESTENV

```
###############################################################################
# Environment variables for running TPC-C
###############################################################################
unset NO_STDERR
unset ABORT

setenv COMMENT   "V2500, 7488-WH DB, Oracle 8.1.5, Giagabit Ethernet connection"
setenv DATABASE   "oracle"       # name of the database used to run the test
                                 # can be either "oracle", "sybase", or
                                 # "sqlserver"
setenv OPS                0      # Set to 1 if using OPS
setenv ccNUMA             0      # Set to 1 if running on a ccNUMA system
setenv NT                 0      # Set to 1 if using NT
setenv BATCH_TPCC         0      # Set to 1 for "batch_tpcc" with the
                                 # RUNME interface, 0 for c/s TPC-C.
setenv TESTROOT     results
setenv RESULTS_NAME audit          # Directory name of RESULTS (put in root of
                                 # ~tpcc).  So actual directory is
                                 # ~tpcc/${TESTROOT}

setenv TRANS_TIME         80     # Total time to run the test for (in minutes)
setenv SWITCHLOG      1        # Force a switchlog at the begining and end of the
run
setenv CHKPNT_INTERVAL   60000           # Seconds to wait before forcing a
checkpoint
setenv CHKPNT_INTERVAL2  1500 #25' # Seconds to wait before forcing the second

setenv DB_SIZE           7488    # Database size on SUT (<= size actually built)
                                 # value in warehouses


setenv SERVER       "sut2"       # The SUT (Database Server)

setenv CLIENT       "client"     # NOTE:  the client name needs to have a
                                 # suffix of 1,2,3,4,.. etc starting with
                                 # 1 and going to the number of clients.  The
                                 # actual client names will be client1,
                                 # client2, client3, etc.  You need to put
                                 # the base client name here.
setenv NR_CLIENT    "24"          # number of clients

setenv DRIVER       "driver"     # NOTE:  the driver name needs to have a
                                 # suffix of 1,2,3,4,.. etc starting with
                                 # 1 and going to the number of drivers.  The
                                 # actual driver names will be driver1,
                                 # driver2, driver3, etc.  You need to put
                                 # the base driver name here.
setenv NR_DRIVER        "12"      # number of drivers
setenv NR_HOSES         "1"      # For multiple lans between client and server
setenv NR_LAN           "1"      # For multiple lans between driver and client

#
# statistics, should probably all be off during your audit runs (performance
# runs).
#
setenv DATABASE_SNAP     1      # collect database statistics
setenv NUMPROC                 0      # monitor processes on all the system

#
# Audit related stuff + misc
#
setenv RUNCHECK          1
setenv CONSISTENCY       1       # run consistency checks before/after run
                                 # this should be 1 when doing your final
                                 # performance runs.
setenv OUTPUT_LEVEL      3       # minimum level - 3
                                 # maximum level - 1
                                 # need to set to 1 for durability tests

setenv REMOVE_OUTPUT     0       # set to 1 to compress "success" and
                                 # "deliv_results" files after each run
setenv COMPRESS_OUTPUT   0       # set to 1 to compress "success" and
                                 # "deliv_results" files after each run

setenv CONFIG_FILE  /O/bench/tpc/tpcc/admin/p_run.ora  # database configuration file

###############################################################################
# The lines below should really not be modified much (if at all)
###############################################################################
# For ODBC
# setenv SHLIB_PATH /opt/odbc/drivers:/opt/odbc/lib
setenv TRANS_NUM   1300000000      # Total number of transactions to run

setenv DELIVERY_LOGS  logs       # Directory name for logfiles

setenv RPT_WINDOW_SIZE   30      # Reporting window size in number of
                                       # RPT_GRANULARITY; for example,
                                       # window size is 10 minutes if
                                       # RPT_GRANULARITY=30 and RPT_WINDOW_SIZE=20

setenv TRANS_TYPE        0       # 0=all, 1=new-order, 2=payment,
                                 # 3=order_status, 4=delivery, 5=stock_level


#
# For TPC-C rev 3.1 and later the difference between the LOAD value of
# CLAST_CONST_C and the run value needs to be within 65-119 inclusive
# but can't be 96 or 112
#
setenv CLAST_CONST_C     86    # a run-time constant chosen within [0..255]
setenv CID_CONST_C       498    # a run-time constant chosen within [0..1023]
setenv IID_CONST_C      3415    # a run-time constant chosen within [0..8191]

setenv COPY_ENV              1          # 1 = Copy TESTENV to other Drivers.
                                    # 0 = DO NOT copy. It is the tester's
                                        #     responsibility to make TESTENVs on all
                                        #     the other drivers.

setenv COMM_ADJUST_NEWO        0.00   # new-order comm delay, Convert TELNET to DTCs
setenv COMM_ADJUST_PMT         0.00   # payment comm delay
setenv COMM_ADJUST_ORDS        0.00   # order-status comm delay
setenv COMM_ADJUST_DVRY        0.00     # delivery comm delay
setenv COMM_ADJUST_STKL        0.00     # stock-level comm delay

setenv NEWO_MENU         0.00   # new order menu RTE delay
setenv PMT_MENU          0.00   # payment menu RTE delay
setenv OS_MENU           0.00   # order status menu RTE delay
setenv DVRY_MENU         0.00   # delivery menu RTE delay
setenv STKL_MENU         0.00   # stock menu RTE delay

#
# Keying times  Don't change these unless doing special tests.  They need
# to be float values.
#
setenv NEWO_KEY          18.01   # new order keying time (18.0)
setenv PMT_KEY            3.01   # payment keying time (3.0)
setenv OS_KEY             2.01   # order status key time (2.0)
setenv DVRY_KEY           2.01   # delivery key time (2.0)
setenv STKL_KEY           2.01   # stock level key time (2.0)

#
# Think times.  Twiddle these as needed.  They need to be float values.
#
```

```
setenv NEWO_THINK        12.12   # new order keying time (12.20)
setenv PMT_THINK         12.05   # payment keying time (12.20)
setenv OS_THINK          10.10   # os keying time (10.25)
setenv DVRY_THINK         5.05   # delivery keying time (5.20)
setenv STKL_THINK         5.05   # stock level keying time (5.20)

setenv RANDOMIZE_OUTPUT      1   # Specifies the percentage of users that should
                                 # output full terminal data (the works) even
                                 # if the OUTPUT_LEVEL is not at 1
```

D.2 Field Value Generation

Source/src/driver/generate.c

```
/*****************************************************************************
 @(#) Version: A.10.10 $Date: 97/12/15 13:53:51 $

 (c) Copyright 1996, Hewlett-Packard Company, all rights reserved.
*****************************************************************************/

#include <stdio.h>
#include <values.h>
#include <unistd.h>
#include <time.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <fcntl.h>
#include <signal.h>
#include <math.h>

#include "shm_lookup.h"
#include "random.h"

#include <time.h>

int CLAST_CONST_C = 208;
int CID_CONST_C =  37;
int IID_CONST_C = 75;

int trans_type = 0;   /* type of transaction 0 == all */

extern ID warehouse;
extern ID district;

extern int no_warehouse;
extern int no_item;
extern int no_dist_pw;
extern int no_cust_pd;
extern int no_ord_pd;
extern int no_new_pd;
extern int tpcc_load_seed;

neworder_gen(t)
    neworder_trans *t;
    {
    int i;

    t->W_ID = warehouse;

    t->D_ID = RandomNumber(1, no_dist_pw);
    t->C_ID = NURandomNumber( 1023, 1, no_cust_pd, CID_CONST_C);

    t->O_OL_CNT = RandomNumber(5, 15);

    for (i=0; i<t->O_OL_CNT; i++)
        {
        t->item[i].OL_I_ID = NURandomNumber(8191, 1, no_item, IID_CONST_C);
        t->item[i].OL_SUPPLY_W_ID = RandomWarehouse(warehouse, scale, 1);
        t->item[i].OL_QUANTITY = RandomNumber(1, 10);
        }

    /* 1% of transactions roll back. Give the last order line a bad item */
    if (RandomNumber(1, 100) == 1)
        t->item[t->O_OL_CNT - 1].OL_I_ID = -1;
    }

payment_gen(t)
    payment_trans *t;
    {

    /* home warehouse is fixed */
    t->W_ID = warehouse;

    /* Random district */
    t->D_ID = RandomNumber(1, no_dist_pw);

    /* Customer is from remote warehouse and district 15% of the time */
    t->C_W_ID = RandomWarehouse(warehouse, scale, 15);
    if (t->C_W_ID == t->W_ID)
        t->C_D_ID = t->D_ID;
    else
        t->C_D_ID = RandomNumber(1, no_dist_pw);

    /* by name 60% of the time */
    t->byname = RandomNumber(1, 100) <= 60;
    if (t->byname)
        LastName(NURandomNumber(255, 0, no_cust_pd/3 - 1, CLAST_CONST_C),
                 t->C_LAST);
    else
        t->C_ID = NURandomNumber(1023, 1, no_cust_pd, CID_CONST_C);

    /* amount is random from [1.00..5,000.00] */
    t->H_AMOUNT = RandomNumber(100, 500000);

    }

ordstat_gen(t)
    ordstat_trans *t;
    {

    /* home warehouse is fixed */
    t->W_ID = warehouse;

    /* district is randomly selected from warehouse */
    t->D_ID = RandomNumber(1, no_dist_pw);

    /* by name 60% of the time */
    t->byname = RandomNumber(1, 100) <= 60;
    if (t->byname)
        LastName(NURandomNumber(255, 0, no_cust_pd/3 - 1, CLAST_CONST_C),
                 t->C_LAST);
    else
        t->C_ID = NURandomNumber(1023, 1, no_cust_pd, CID_CONST_C);
    }

delivery_gen(t)
    delivery_trans *t;
    {
    t->W_ID = warehouse;
    t->O_CARRIER_ID = RandomNumber(1,10);
    }

stocklev_gen(t)
    stocklev_trans *t;
    {
    t->W_ID = warehouse;
    t->D_ID = district;
    t->threshold = RandomNumber(10, 20);
    }

int get_trans_type()
/*****************************************************************************
 * get_trans_type selects a transaction according to the weighted average
 * For TPC-C rev 3.0 and less and TPC-C rev 3.2 this is:
 *       new-order :  ???
 *       payment   : 43.0%
 *       order stat:  4.0%
 *       delivery  :  4.0%
 *       stock     :  4.0%
 *****************************************************************************/
```

```
    {
    static double weight[] = { 0.0,  0.0, .4305, .0405, .0405, .0405};
    double drand48();
    int type;
    double r;

    /* choose a random number between 0.0 and 1.0 */
    if (trans_type == 0) {
#ifdef USE_DRAND48
        r = drand48();
#else
        r = randy();
#endif

        /*
         * select one of STOCKLEV, DELIVERY, ORDSTAT and PAYMENT
         * based on weight
         */
        for (type = STOCKLEV;  type > NEWORDER;  type--) {
            r -= weight[type];
            if (r < 0) break;
        }
    } else {
          /* user wants only a certain type (say all stocklevel) so do that
           instead */
          type = trans_type;
    }
    /* return the value of the selected card, or NEWORDER if none selected */
    return type;
    }
```

Appendix E Disk Storage

The calculations used to determine the storage requirements for the 8 hours logical log and the 180-day space calculations are contained in this appendix.

The calculations for the 8 hours recovery log were based on the Oracle statistic "redo blocks written" per user commit multiplied by the number of commits per tpmC. Datat was gathered from the measurement interval of Run1 with full terminal load (74880 users).

For a measurement run, we computed the logspace as follows:

The number of redo blocks per tpmC:

(for this run tpmC = 92832.96)

During the measurement interval:
---

number of new order transactions = 2413657

number of rolled back new order transactions = 24175

number of commited new order transactions = 2413657 - 24175 = 2389482

number of payment transactions = 2320492

number of delivery transactions = 218272

user commits per tpmC = (2389482 + 2320492 + 218272) / 2413657 = 2.0418

Note: The Order-Status and Stock-Level transactions commit but are read only transactions and thus are not accounted for in the 'user commit' Oracle statistic).

Oracle statistic for entire run duration:
---

redo blocks written (total) = 107678575 (from Oracle stats)

redo block size = 1024 bytes

user commits (total) = 14533038

redo blocks written per user commit = 107678575 / 14533038 = 7.4092

Log space per tpmC:
---

redo space/min/tpmC = 7.4092 * 2.0418 = 15.1281 KB

8 hours log space = 15.1281 * 92832.96 * 8 * 60 = 642.877 GB (must be mirrored)

On the 10 log disk arrays, we have configured 644.22 GB of space

During the runs, the log space was configured on the 10 log arrays as follow:

LUN #1:19,968MB
LUN #2: 46,000MB

All 10 arrays in the following list were identically configured:
/dev/rdsk/c35t0d0
/dev/rdsk/c13t0d0
/dev/rdsk/c18t0d0
/dev/rdsk/c42t0d0
/dev/rdsk/c16t0d0
/dev/rdsk/c8t0d0
/dev/rdsk/c22t0d0
/dev/rdsk/c47t0d0
/dev/rdsk/c48t0d0
/dev/rdsk/c9t0d0

Total log space allocated: 644.22 GB

The log1 and log2 logical volumes were striped evenly across the 10 arrays with a stripe size of 64KB. With a redo log space usage of 15.1281KB per tpmC, after every 42-43 TPC-C Business Transactions, a block would be written to each of the 10 arrays. This even usage pattern continues for the full 8-hour period.

TSPACE NAME	BLOCKS ALLOCATED	INITIAL STATIC	INITIAL DYNAMIC	8-HOUR REQUIRED	OVERSIZE (BLOCKS)
CUST	118,652,928	117,936,001	0	117,936,001	716,927
HIST	10,242,048	0	6,732,806	8,068,332	2,173,716
ICUST1	4,096,512	3,225,600	0	3,225,600	870,912
ICUST2	8,193,024	6,021,120	0	6,021,120	2,171,904
IORD1	5,633,024	4,032,000	0	4,032,000	1,601,024
IORD2	8,193,024	4,300,800	0	4,300,800	3,892,224
ISTK	15,362,048	9,596,160	0	9,596,160	5,765,888
NORD	1,280,512	672,000	0	672,000	608,512
ORDL	137,105,920	0	89,303,040	107,017,283	30,088,637
ORDR	7,425,024	0	4,674,451	5,593,360	1,831,664
ROLL	410,112	410,112	0	410,112	0
STOK	157,642,752	157,248,005	0	157,248,005	394,747
SYSTEM	410,112	122,051	0	122,051	288,061

TPC-C 180 Day Space Requirements

The calculations used to determine the storage requirements for the 8 hour logical log and the 180-day space are contained in this appendix.

TPC-C 180 Day Space Requirements

TPM	92832.96
Warehouses	7488

SEGMENT	TYPE	TSPACE	BLOCKS	FIVE_PCT	DAILY_GROW	TOTAL
CUSTOMER	TABLE	CUST	112,320,001	5,616,000	0	117,936,001
DISTRICT	TABLE	SYSTEM	74,887	3,744	0	78,631
HISTORY	TABLE	HIST	6,732,806	0	1,335,526	8,068,332
ICUSTOMER	INDEX	ICUST1	3,072,000	153,600	0	3,225,600
ICUSTOMER2	INDEX	ICUST2	5,734,400	286,720	0	6,021,120
IDISTRICT	INDEX	SYSTEM	1,032	52	0	1,084
IITEM	INDEX	SYSTEM	1,149	57	0	1,206
INORD	INDEX	NORD	640,000	32,000	0	672,000
IORDERS	INDEX	IORD1	3,840,000	192,000	0	4,032,000
IORDERS2	INDEX	IORD2	4,096,000	204,800	0	4,300,800
IORDL	INDEX	ORDL	89,303,040	0	17,714,243	107,017,283
ISTOCK	INDEX	ISTK	9,139,200	456,960	0	9,596,160
ITEM	TABLE	SYSTEM	6,667	333	0	7,000
IWAREHOUSE	INDEX	SYSTEM	85	4	0	89
ORDERS	TABLE	ORD	4,674,451	0	927,229	5,601,680
ROLL_SEG	SYS	ROLL	410,112	0	0	410,112
STOCK	TABLE	STOK	149,760,005	7,488,000	0	157,248,005
SYSTEM	SYS	SYSTEM	26,178	0	0	26,178
WAREHOUSES	TABLE	SYSTEM	7,489	374	0	7,863
Total			389,839,502	14,434,644	19,976,998	424,251,144

Dynamic space	100,710,297	Initial Blocks for (History+Orders+Order_Line)
Static space	303,563,849	Initial Blocks + 5% - Dynamic
Daily Growth	19,976,998	Total Dynamic [(calc. as (Initial Blocks)*tpmC/(WHS*62.5)]
Daily Spread	0	Oracle may be configured so that daily spread is 0
180-day space (blk.)	3,899,423,489	Static + 180*(Daily Growth+Daily Spread)
Block size (bytes)	2,048	
180-day (GB)	7,437.56	Excludes OS, Paging and RDBMS Logs
Log block size (bytes)	1,024	
Log blocks/min/tpmC	15.13	Number of log blocks used per minute per tpmC
8-hour log (GB)	642.88	RDBMS Logs *10 Log disks*
Server swap (GB)	3.97	OS: Paging
Server OS (GB)	3.45	OS: UNIX File System
Total Space Needed	8,087.86	GB

Priced-Sytem Configuration	**Size (MB)*See Note1**	**Quantity**	**Total (GB)**
AutoRAID 12H with 12 4.3 GB disk drives	39,598	18	696.06
AutoRAID 12H with 12 9.1 GB disk drives	86,186	88	7,406.61
Total Storage in Priced System (GB)	**(NOTE1: The sizes given in MB are after RAID5 redundency)**		8,102.67

Appendix F Price Quotes

The following pages contain the price quotes for the hardware included in this FDR.

To: Norbert Boege
Hewlett-Packard Company
19111 Pruneridge Avenue
Cupertino, CA 95014

From: Sandy Potter
Sun Data Corporation
1 Sun Court
Norcross, GA 30092

Date: 2-Mar-99

Item Description	Part Number	Unit Price	Qty	Extended Price	5 yr. Maint. Price
Server Hardware					
HP 9000 V2500 Enterprise Server	A5074A	138600	1	138600	119532
Add'l Dual 440 MHz PARISC 8500 CPU's	A5492A Opt. 0D1	45500	16	728000	118272
System Mgmt. Station	A4802B	8036	1	8036	2116.8
Memory Controller Board	A5078A Opt 0D1	7000	4	28000	
256 MB DIMM for a total of 2 GB	A5082A Opt. 0D1	24500	16	392000	
PCI FWD SCSI-2 Card	A4800A Opt 0D1	840	1	840	
PCI Fibre Channel 1062 Mb/s	A3740A Opt. 0D1	1890	11	20790	
PCI 1000 BaseSX Lan	A4926A Opt 0D1	1466.5	1	1466.5	
PCI 64-port MUX	J3593A Opt. 0D1	1046.5	1	1046.5	
16 port RS-232 DB25 port module	J2485A	616	1	616	
25 ft cable	J3595A	105	1	105	
5.5 kVA UPS	A3589A Opt. 002	7000	1	7000	336
1.8 kVA UPS rackmount	A2997A Opt. 002	2310	53	122430	16027.2
2m Filed Integrated Cabinet	C2787A	1400	31	43400	
FC SCSI Multiplexer	A3511A	7280	11	80080	71018.64
16 bit FW SCSI Adapter	A3511A Opt. 003	906.5	36	32634	
Auto Raid Array Model 12H	A3700AZ	3050	106	323300	560210
Two 96 MB controllers with Auto Raid	A3700AZ Opt. 203	9200	106	975200	
Twelve 9.1 GB disk modules, 10K rpm	A3700AZ Opt. 172	15120	18	272160	
Twelve 4.3 GB disk modules	A3700AZ Opt. 112	10080	18	181440	
Twelve 9.1 GB disk modules	A3700AZ Opt. 132	13860	70	970200	
0.9m 68-pin high density male connec.	A3700AZ Opt. 801	84	70	5880	
2.5m HDTS68 to HDTS68 Cable	C2924A	98	36	3528	
			Subtotal	4336752	887512.64
Server Software					
HP-UX 11 Instant Ignition	B3920EA, Opt 0D1	136.5	1	136.5	
HP-UX 11 Unlimited License for V-Class	B3919EA, Opt B9V	0	1	0	
			Subtotal	136.5	0
Client Hardware					
Hewlett Packard Model C360 Workstation	A4988A	11200	24	268800	75801.6
256 MB ECC memory module	A4517A	1606.5	144	231336	
Console	C1084GX	377.3	1	377.3	168
4 GB ultra SCSI disk	A4569A	700	24	16800	
100 Base T PCI LAN Adapter	B5509BA	241.5	72	17388	
			Subtotal	534701.3	75969.6
Client Software					
HP C/ANSI C Compiler	B3901BA Opt AH0	1102.5	1	1102.5	980
			Subtotal	1102.5	980
User Connectivity					
HP ProCurve Switch 4000M	J4121A	2519.3	1	2519.3	588
HP ProCurve Switch Gigabit-SX Module	J4113A	1049.3	1	1049.3	
HP ProCurve Switch 10/100Base-T Module	J4111A	489.3	1	489.3	
			Subtotal	4057.9	588
			Total	4876750.2	965050.24

Note: Above configuration are generally available and prices are guaranteed for 60 days. The service prices are based on a service agreement for the entire system. The above prices may vary if items are purchased separately. Pricing is subject to approved credit terms.

Sandy Potter

BEA ENTERPRISE MIDDLEWARE SOLUTIONS

March 2, 1999

Ms. Kristin Lichens
Hewlett Packard
Santa Clara, CA

Dear Ms. Lichens :

Per your request I am enclosing the pricing information regarding TUXEDO 6.x that you requested. This pricing applies to Tuxedo 6.1, 6.2, 6.3 and 6.4. Please note that Tuxedo 6.4 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, HP server systems are classified in one of 5 tiers based on CPU type and capacity. The HP V2XXX series is in Tier 5. 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. 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.

BEA Tux/CFS Unlimited User License Fees Per Server

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 (Class 1and Class 2)	Unlimited	$3,000.00	$450.00	$660.00
Tier 2 -- PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations (class 3)	Unlimited	$12,000.00	$1,800.00	$2,640.00
Tier 3 -- Midrange Multiprocessors, up to 8 CPUs per system capacity (Class 4 and 5)	Unlimited	$30,000.00	$4,500.00	$6,600.00
Tier 4 -- Large (more than 8, less than 32 CPUs)	Unlimited	$100,000.00	$15,000.00	$22,000.00

BEA SYSTEMS, INC
2315 NO. FIRST STREET
SAN JOSE, CA 95131

FAX: 408-570-8901
PHONE: 408-570-8000

Tier 5 -- Massively Parallel Systems, > 32 processors	Unlimited	$250,000.00	$37,500.00	$55,000.00

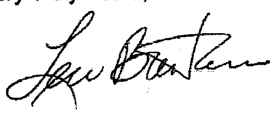
HP 9000 server tier classifications

Platform	Tier 1	Tier 1	Tier 2	Tier 3	Tier 3	Tier 4	Tier 5
Hewlett-Packard	Uni-processor Workstations	9000/E25 9000/E35 9000/E45 9000/E55 9000/G30 9000/G40	9000/G50 9000/G60 Multi-Processor Workstations 9000/D200, 210 220,230	9000/H20 9000/H30 9000/H40 9000/H50 9000/I30 9000/I40 9000/K1XX 9000/D310, 320,330 9000/D250/60	9000/I50,60 9000/H60 9000/G70 9000/H70 9000/I70 9000/K2XX 9000/K3XX 9000/K4XX 9000/K5XX 9000/D350/60/70	9000/T500, T600 1-16 CPUs	9000/V series all models

Intel based server tier classifications:

Platform	Operating System	Tier 1	Tier 1	Tier 2	Tier 3	Tier 3
Intel Pentium/ Pentium Pro PCs	Interactive R3.2 ESIX SVR 4.0 SCO UNIX 3.2.2 and 3.2.4 SCO ODT 2.x,3.x Solaris x86 2.X UnixWare, Windows NT 3.5/4.0	All 386/486 PCs are Class 1	ALL Pentium and Pentium Pro PCs with 1 or 2 CPUs capacity are Tier 1	ALL Pentium and Pentium Pro PCs with 3 or 4 CPUs capacity are Tier 2		ALL Pentium and Pentium Pro PCs with 5,6,7, or 8 CPUs are Tier 3

Very Truly Yours,



Lewis D. Brentano,
Director, Market Planning

Software House International
Pricing Proposal

Quotation #MO-990301-56800
03/01/99

Hewlett Packard
Norbert Boege
Quote Good for Sixty Days

Phone: **Fax: 408-447-4594**

SHI Account Exec: Matthew O. Martin
Telephone : (408) 922-1106
Fax : (408) 526-1222

Reference:

Product	Part #	Qty	List	Your Price	Total
8Port RJ45 Hub	Z79445	1		$29.00	$29.00
Total					**$29.00**

Additional Comments: