Hewlett-Packard Company

TPC Benchmark™ H
Full Disclosure Report

HP ProLiant DL740 Cluster 32P
using
Oracle Database 10g Enterprise Edition with
Real Application Cluster and
Partitioning; and
Red Hat Enterprise Linux AS 3

First Edition
March 2004

First Edition – March 2004

Hewlett Packard Company, the Sponsor of this benchmark test, 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. The Sponsor 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, the Sponsor 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, the TPC Benchmark H 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. No warranty of system performance or price/performance is expressed or implied in this report.

HP ProLiant is a registered trademark of Hewlett Packard Company.

Oracle Database 10g Enterprise Edition with Real Application Cluster and Partitioning are trademarks of Oracle Corporation

TPC Benchmark, TPC-H, QppH, QthH and QphH are trademarks of the Transaction Processing Performance Council.

All other brand or product names mentioned herein must be considered trademarks or registered trademarks of their respective owners.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on the HP ProLiant DL740 Cluster 32P using Oracle Database 10g Enterprise Edition with Real Application Cluster and Partitioning, in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 2.1.0. The operating system used for the benchmark was Red Hat Enterprise Linux AS 3.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	QppH @ 3TB	QthH @ 3TB	QphH @ 3TB	$ / QphH @ 3TB
HP ProLiant DL740 Cluster 32P	Oracle Database 10g Enterprise Edition with Real Application Cluster and Partitioning, and Red Hat Enterprise Linux AS 3	$2,076,265	30430.7	16470.8	22387.9	$93

The TPC Benchmark™ H was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.

Standard and Executive Summary Statements

Executive Summary and Numerical Quantities Summary of the benchmark results for the HP ProLiant DL740 Cluster 32P can be found in the following pages.

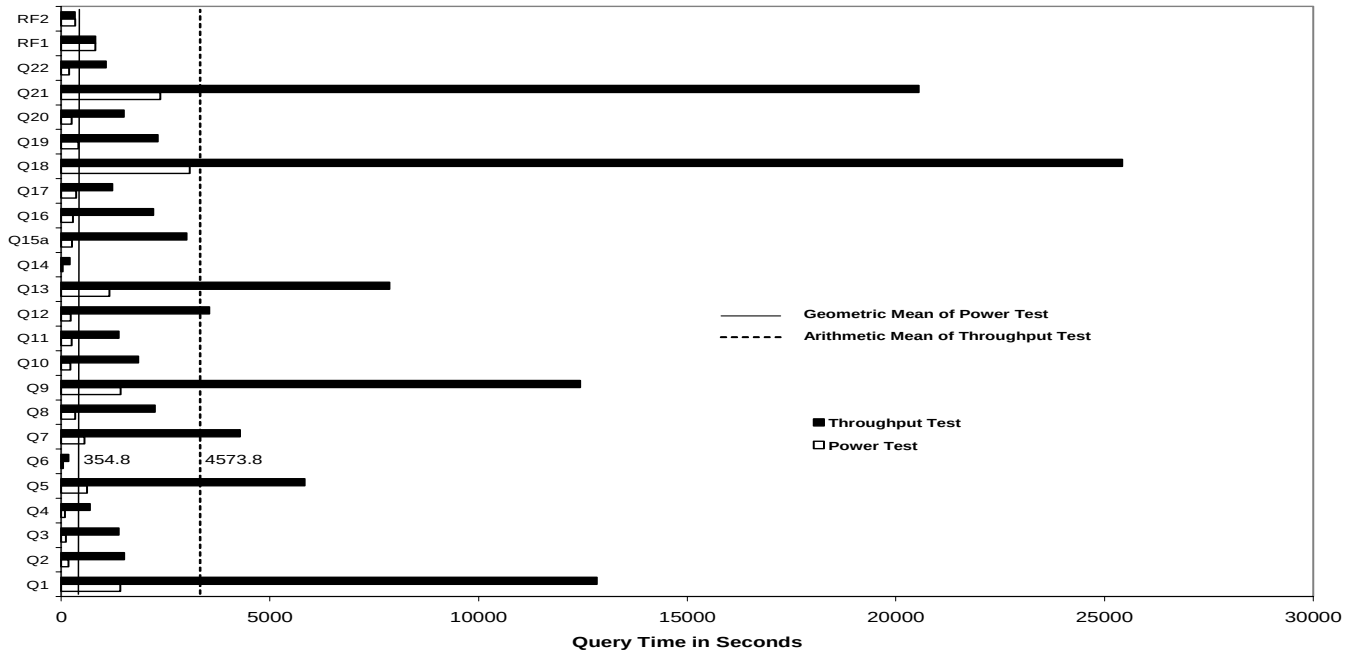
Auditor

The benchmark configuration, environment and methodology were audited by Lorna Livingtree of Performance Metrics Inc. to verify compliance with the relevant TPC specifications.

invent	**HP ProLiant DL740 Cluster 32P**	TPC-H Rev. 2.1.0 Report Date: **March 2, 2004**

Total System Cost	Composite Query per Hour Metric	Price / Performance
$2,076,265	**22387.9** QphH@3000GB	**$93** $ / QphH@3000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
3000GB	**Oracle Database 10g Enterprise Edition with Real Application Cluster and Partitioning**	**Red Hat Enterprise Linux AS 3**		**March 2, 2004**



Database Load Time = 18:52:00	Load Included Backup: N	Total Data Storage / Database Size = 10.75
RAID (Base tables only): Y	RAID (Base tables and auxiliary data structures): Y	RAID (All): Y

System Configuration :	8 nodes
Processors (per node) :	4 x 3.0GHz Intel Xeon Processor MP w/ 4MB cache
Memory (per node) :	8 GB
OS Disk Drives (per node) :	2 x 36GB 15krpm HDD Ultra320
NICs (per node) :	2 x on-board, 1 x hp NC7770 PCI-x Gigabit server adapter
Disk Controllers (per node) :	3 x hp StorageWorks fca 2214DC, 2 x hp StorageWorks fca 2214
Storage Area Network :	8 x hp StorageWorks SAN Switch 2/16 64 x hp StorageWorks MSA1000 896 x 36GB 15krpm HDD Ultra320
Total Storage :	32256GB
Cluster Interconnect :	2 x hp ProCurve Switch 4148gl

	HP ProLiant DL740 Cluster 32P				TPC-H Rev. 2.1.0		
					Report Date:		2-Mar-04
Description	**Part Number**	**Third Party**		**Unit Price**	**Qty**	**Extended Price**	**3 yr. Maint. Price**
Server Hardware		**Brand**	**Pricing**				
DL740 X3.0-4M 4P 4G ALL	348445-B21		1	40,088	8	320,704	
512MB 133MHz ECC SDRAM Memory Option Kit (1x512MB)	236853-B21		1	309	64	19,776	
36GB, 15krpm HDD Ultra320 HP	286776-B22		1	429	16	6,864	
S5500 15" Monitor	261602-001		1	129	8	1,032	
Scroll Mouse-Carbon	231947-B21		1	5	8	40	
PS/2 Easy Access Internet Keyboard	265977-001		1	12	8	96	
hp NC7770 PCI-X Gigabit server adapter	244948-B21		1	221	8	1,768	
hp StorageWorks fca 2214 2Gb, 64-bit/133Mhz PCI-X FC HBA	281541-B21		1	1,590	16	25,440	
hp StorageWorks fca 2214DC 2Gb, 64-bit/133Mhz PCI-X FC HBA	321835-B21		1	2,500	24	60,000	
Servers 500 Series CarePaq, 3yr 7x24, 4hr resp.	401782-002		1	1,795	8		14,360
Rack Model 9142 (42U - Opal) - Flat Pallet	120663-B21		1	1,321	8	10,568	
Rack Sidewall Kit	120670-B21		1	207	1	207	
UPS T1500 XR	204155-002		1	621	8	4,968	
hp ProCurve Switch 4148gl	J4888A		1	1,934	3	5,802	
hp ProCurve Switch gl 100/1000-T module	J4863A		1	812	12	9,744	
hp ProCurve Switch gl 100/1000-T module Support 3YR 24X7 4HR	U2856E		1	1,080	3		3,240
					Subtotal	**467,009**	**17,600**
Storage							
hp Storageworks Modular SAN Array 1000	201723-B22		1	9,995	64	639,680	
hp StorageWorks Modular SAN Array 1000 Support 3YR 24x7 4HR	402164-002		1	3,538	64		226,432
hp StorageWorks SAN switch 2/16-EL	322120-B21		1	6,400	8	51,200	
hp SAN Switch 2/32 Support 3YR 24X7 4HR	161299-002		1	3,399	8	27,192	
5m LC to LC Cable Kit	221692-B22		1	82	32	2,624	
15m LC to LC Cable Kit	221692-B23		1	103	32	3,296	
2Gb SFF-SW Trncvr Kit	221470-B21		1	199	64	12,736	
2Gb SFF-SW Trncvr Kit (10% spares)	221470-B21		1	199	7	1,393	
36GB, 15krpm HDD Ultra320 HP	286776-B22		1	429	896	384,384	
36GB, 15krpm HDD Ultra320 HP (10% spares)	286776-B22		1	429	90	38,610	
					Subtotal	**1,161,115**	**226,432**
Software							
Oracle Database 10g Enterprise Edition for 3 years, Named User Plus	run-time		2	10,000	32	320,000	
Real Application Clusters for 3 years, Named User Plus	run-time		2	5,000	32	160,000	
Partitioning for 3 years, Named User Plus	run-time		2	2,500	32	80,000	
Database Server Support Package (8 servers, 3 years)	run-time		2	16,000	3		48,000
Red Hat Enterprise Linux AS (Ver. 3 Std. Edi. For x86)	SVC0022		3	1,499	8	11,992	
2 Addi. Yrs Subs. to Red Hat Enterprise Linux AS (Ver. 3 Std. Edi. For x86)	SVC0022		3	1,499	16		23,984
					Subtotal	**571,992**	**71,984**
*hp Large Purchase discount			1			(276,781)	(41,485)
Oracle Mandatory E-Business Discount			2			(121,600)	
					Total	**1,801,735**	**274,531**
					3-Year Cost of Ownership:		**$2,076,265**
					QphH Rating:		**22387.9**
Price Key: 1-HP at 17% discount, 2-Oracle, 3-Red Hat. Oracle pricing contact: Mary Beth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081. *All discounts are based on US list prices and for similar quantities and configurations.					**$ / QphH@3000GB:**		**$93**

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

Numerical Quantities

Measurement Results:

Database Scale Factor	= 3000
Total Data Storage / Database Size	= 10.75
Start of Database Load	= 1/16/2004 19:39:12
End of Database Load	= 1/17/2004 14:31:12
Database Load Time	= 18:52:00
Query Streams for Throughput Test	= 8
TPC-H Power	= 30430.7
TPC-H Throughput	= 16470.8
TPC-H Composite Query-per-Hour Metric (QphH@3000GB)	= 22387.9
Total System Price Over 5 Years	= $2,070,568
TPC-H Price/ Performance Metric ($/QphH@3000GB)	= $93

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 115404 seconds

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	Stop Date	Stop Time	Duration
Stream00	117143112	17-Jan-04	15:49:56	17-Jan-04	20:01:34	4:11:38
Stream01	117143113	17-Jan-04	20:01:48	19-Jan-04	01:21:52	29:20:04
Stream02	117143114	17-Jan-04	20:01:48	19-Jan-04	00:17:30	28:15:42
Stream03	117143115	17-Jan-04	20:01:48	18-Jan-04	23:57:11	27:55:23
Stream04	117143116	17-Jan-04	20:01:48	19-Jan-04	00:44:15	28:42:27
Stream05	117143117	17-Jan-04	20:01:48	19-Jan-04	01:32:12	29:30:24
Stream06	117143118	17-Jan-04	20:01:48	18-Jan-04	23:35:24	27:33:36
Stream07	117143119	17-Jan-04	20:01:48	18-Jan-04	19:30:19	23:28:31
Stream08	117143120	17-Jan-04	20:01:48	19-Jan-04	00:52:09	28:50:21
Refresh		19-Jan-04	01:32:12	19-Jan-04	04:05:12	2:33:00

HP ProLiant DL740 Cluster 32P

TPC-H Rev. 2.1.0

Report Date:
March 2, 2004

TPC-H Timing Intervals (in seconds)

Query	1	2	3	4	5	6	7	8
Stream 00	1417.4	180.4	114.2	95.8	622.6	49.1	559.8	343.4
Stream 01	4985.1	1365.2	883.9	166	4440.1	179.2	3572	1599
Stream 02	8868.8	1389.3	651.1	565.2	4392.7	114.7	4285.5	2004.4
Stream 03	7281.5	961.5	517.9	592.3	5831.2	72	2740.2	2248.7
Stream 04	7207.4	1099.5	247.1	696.2	3882.1	171	1690.2	1467.4
Stream 05	12834.9	1515	980.7	468.7	4382.1	70.5	4154.8	1522
Stream 06	10173.4	944.3	1383.1	551	4278.2	72.8	2791.9	1795.1
Stream 07	9303.7	1208.1	740.9	695.1	4642.1	165.7	3309.3	1699.2
Stream 08	12119	1079.4	640.4	487.3	4194.8	164.7	2997.7	1395.4
Min Qi	4985.1	944.3	247.1	166	3882.1	70.5	1690.2	1395.4
Max Qi	12834.9	1515	1383.1	696.2	5831.2	179.2	4285.5	2248.7
Avg Qi	9096.7	1195.3	755.6	527.7	4505.4	126.3	3192.7	1716.4
Query	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Stream 00	1431.5	221.9	250.6	225.3	1157.4	42.1	266.0	281.1
Stream 01	8742.5	1488.2	1114.7	2285.5	8571.4	114.4	3,430.4	2515.5
Stream 02	8466.2	1103.4	1130.7	1679.8	7808.9	250.2	3,036.2	3274.4
Stream 03	10914.4	1205.2	1329.5	2711.7	6963.7	219.3	2,221.5	1681
Stream 04	11233.8	1583.1	1178.5	2421.7	5983.3	236.8	3,116.1	1822.8
Stream 05	2223.4	814.2	1383.9	1400.2	9026.5	211.7	2,163.9	2389.9
Stream 06	9909.6	1218.4	1235.7	3552.3	8101.2	192.1	4,368.7	2052.1
Stream 07	9144	1846.7	1145.6	3272.8	8314.5	186.9	3,028.6	2238.8
Stream 08	12434.8	1298.2	492.7	3100.9	8148.4	207.3	2,710.9	1705.5
Min Qi	2223.4	814.2	492.7	1400.2	5983.3	114.4	2163.9	1681
Max Qi	12434.8	1846.7	1383.9	3552.3	9026.5	250.2	4368.7	3274.4
Avg Qi	9133.6	1319.7	1126.4	2553.1	7864.7	202.3	3009.5	2210.0
Query	**17**	**18**	**19**	**20**	**21**	**22**	**RF1**	**RF2**
Stream 00	361.9	3083.2	411.4	250.1	2376.2	194.2	821.0	341.3
Stream 01	1364.1	27957.9	1083.9	1216.1	27820.7	708.6	777.59	329.67
Stream 02	3376.6	27334.1	3559.9	1010.1	16168.4	1272.3	787.02	311.23
Stream 03	696	31019.4	872	850.7	18203.8	1390.3	793.74	327.62
Stream 04	804	28910.9	2715.3	291.7	26150.8	438.4	761.11	342.06
Stream 05	742.9	27961	2964.3	2528.6	25172.5	1313.4	805.21	322.76
Stream 06	1448.7	25067.9	1755.1	994.6	16260.2	1070.3	924.33	327.7
Stream 07	678.5	8446.9	1868	4334.3	16882.6	1358.5	842.69	346.81
Stream 08	709.8	26699.6	3709	789.6	17739.2	996.6	855.81	324.1
Min Qi	678.5	8446.9	872	291.7	16168.4	438.4	761.11	311.23
Max Qi	3376.6	31019.4	3709	4334.3	27820.7	1390.3	924.33	346.81
Avg Qi	1227.6	25424.7	2315.9	1502.0	20549.8	1068.6	818.4	329.0

Table Of Contents

1.0 General Items

1.1 Test Sponsor

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

Hewlett Packard Company sponsored this benchmark. The benchmark was developed and engineered by Hewlett Packard Company and Oracle Corporation. Testing took place at HP Database Performance Engineering Laboratory in Houston, Texas.

1.2 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 by not limited to:

- *Database Tuning Options*
- *Optimizer/Query execution options*
- *Query processing tool/language configuration parameters*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and configuration parameters*
- *Configuration parameters and options for any other software component incorporated into the pricing structure*
- *Compiler optimization options*

This requirement can be satisfied by providing a full list of all parameters and options, as long as all those which have been modified from their default values have been clearly identified and these parameters and options are only set once.

Appendix A contains Database and Operating system configuration and tunable parameters.

1.3 Configuration Items

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 their protocol type.*
- *Number of LAN (e.g. Ethernet) Connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.*
- *Type and the run-time execution location of software components (e.g., DBMS, query processing tools/languages, middle-ware components, software drivers, etc.).*

The hp ProLiant DL740 Cluster 32P is an eight node cluster of hp ProLiant DL740 servers, depicted in Figure 1.1, consisted of:

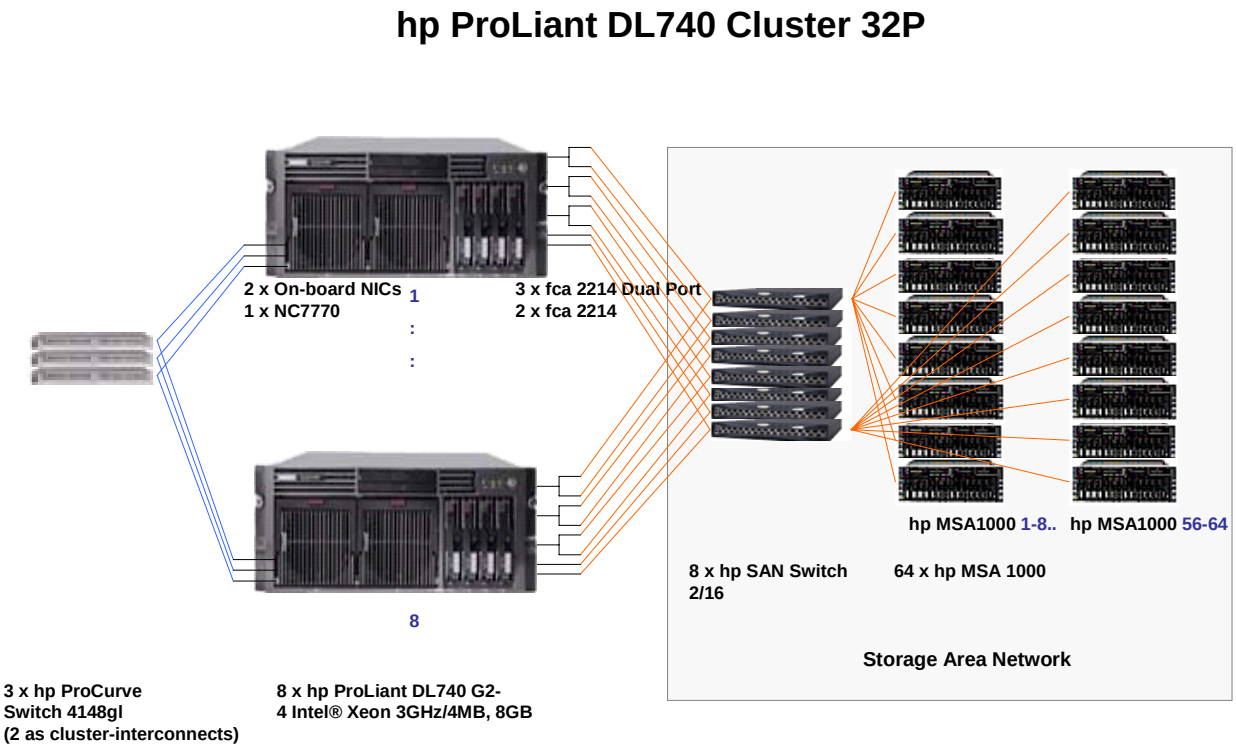
Processors (per node) :	4 x 3.0GHz Intel Xeon Processor MP w/ 4MB cache
Memory (per node) :	8 GB
OS Disk Drives (per node) :	2 x 36GB 15krpm HDD Ultra320
NICs (per node) :	2 x on-board (used as cluster interconnects)
	1 x hp NC7770 PCI-x Gigabit server adapter
Disk Controllers (per node) :	3 x hp StorageWorks fca 2214DC, 2 x hp StorageWorks fca 2214
Storage Area Network :	8 x hp StorageWorks SAN Switch 2/16
	64 x hp StorageWorks MSA1000
	896 x 36GB 15krpm HDD Ultra320
Total Storage :	32256GB
Cluster Interconnect :	2 x hp ProCurve Switch 4148gl

The storage area network (SAN) consisted of eight hp SAN Switch 2/16s and 64 hp StorageWorks MSA1000s. The hp ProLiant DL740 servers had three hp StorageWorks fca 2214DC (dual port) HBAs and two StorageWorks fca 2214 (single port) HBAs; connected to hp SAN Switch 2/16s. Each hp SAN Switch 2/16s had eight hp StorageWorks MSA1000s connected to it. Each hp StorageWorks MSA1000 had one RAID0+1 volume. The array accelerator cache was set to 100% read. All tables except NATION and REGION had 64 horizontal partitions; each of them resided on one of the 64 hp StorageWorks MSA1000s. The temp files and flatfiles were also distributed across the 64 hp StorageWorks MSA1000s. Eight of the hp StorageWorks MSA1000s had two additional partitions for database redo log files.

A detailed description of distribution of database files can be found in Table 5.2.

The cluster interconnects were two to hp ProCurve Switch 4148gls, connected to the on-board NICs.

Figure 1.1: Benchmarked and Priced Configuration



2.0 Clause 1: Logical Database Design

2.1 Database Definition Statements

Listings must be provided for all table definition statements and all other statements used to set up the test and qualification databases. (8.1.2.1)

Appendix B contains the database build scripts.

2.2 Physical Organization of Database

The physical organization of tables and indices, within the test and qualification databases, must be disclosed. If the column ordering of any table is different from that specified in Clause 1.4, it must be noted.

Please refer Appendix B for column reordering of tables.

2.3 Horizontal Partitioning

Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed.

Horizontal partitioning was used for all tables except NATION and REGION as described in Appendix B.

2.4 Replication

Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.6.

The database was not replicated.

3.0 Clause 2: Queries and Refresh Functions Related Items

3.1 Query Language

The query language used to implement the queries must be identified.

SQL was the query language used to implement all queries.

3.2 Random Number Generation

The method of verification for the random number generation must be described unless the supplied DBGEN and QGEN were used.

TPC supplied versions 1.3.0 of DBGEN and QGEN were used for this TPC-H benchmark.

3.3 Substitution Parameters Generation

The method used to generate values for substitution parameters must be disclosed. If QGEN is not used for this purpose, then the source code of any non-commercial tool used must be disclosed. If QGEN is used, the version number, release number, modification number and patch level of QGEN must be disclosed.

The supplied QGEN version 1.3.0 was used to generate the substitution parameters.

3.4 Query Text and Output Data from Database

The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. If minor modifications (see Clause 2.2.3) have been applied to any functional query definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request..

Appendix D contains the query text and output.

3.5 Query Substitution Parameters and Seeds Used

All the query substitution parameters used during the performance test must be disclosed in tabular format, along with the seeds used to generate these parameters.

Appendix E contains the query substitution parameters and seed used.

3.6 Isolation Level

The isolation level used to run the queries must be disclosed. If the isolation level does not map closely to one of the isolation levels defined in Clause 3.4, additional descriptive detail must be provided.

The queries and transactions were run with the isolation level set to “Level 3” (repeatable read).

3.7 Refresh Functions

The details of how the refresh functions were implemented must be disclosed (including source code of any non-commercial program used).

The refresh function is part of the implementation-specific layer/driver code included in Appendix F.

4.0 Clause 3: Database System Properties

4.1 Atomicity Requirements

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing the code written to implement the ACID Transaction and Query.

Appendix C contains the source code for the ACID transactions.

4.1.1 Atomicity of the Completed Transactions

Perform the ACID Transaction for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDER, LINEITEM, and HISTORY tables.

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

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1.
3. The ACID Transaction committed.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had been changed.

4.1.2 Atomicity of Aborted Transactions

Perform the ACID transaction for a randomly selected set of input data, submitting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDER, LINEITEM, and HISTORY tables.

The following steps were performed to verify the Atomicity of the aborted ACID transactions:

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1. The transaction was stopped prior to the commit.
3. The ACID Transaction was ROLLED BACK.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had not been changed.

4.2 Consistency Requirements

Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another.

A consistent state for the TPC-H database is defined to exist when:

*O_TOTALPRICE = SUM(L_EXTENDEDPRICE – L_DISCOUNT) * (1 + L_TAX)*
For each ORDER and LINEITEM defined by (O_ORDERKEY = L_ORDERKEY)

The following queries were executed before and after a measurement to show that the database was always in a consistent state both initially and after a measurement.

SELECT DECIMAL (SUM (DECIMAL (INTEGER (INTEGER (DECIMAL (INTEGER (100 * DECIMAL (L_EXTENDEDPRICE, 20, 3)), 20, 3) * (1 – L_DISCOUNT)) * (1 + L_TAX)), 20, 3) / 100.0) 20, 3) FROM TPCD.LINEITEM WHERE L_ORDERKEY = okey

SELECT DECIMAL(SUM(O_TOTALPRICE, 20, 3)) from TPCH.ORDERS WHERE O_ORDERKEY = okey

4.2.1 Consistency Tests

Verify that ORDER and LINEITEM tables are initially consistent as defined in Clause 3.3.2.1, based upon a random sample of at least 10 distinct values of O_ORDERKEY.

The following steps were performed to verify the Consistency of ACID transactions:

1. The consistency of the ORDERS and LINEITEM tables was verified based on a sample of order keys.
2. 100 ACID Transactions were submitted from each of 8 execution streams.
3. The consistency of the ORDERS and LINEITEM tables was re-verified.

4.3 Isolation Requirements

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.

4.3.1 Isolation Test 1 - Read-Write Conflict with Commit

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is committed.

The following steps were performed to satisfy the test of isolation for a read-only and a read-write committed transaction:

1. An ACID Transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID Transaction was suspended prior to COMMIT.
2. An ACID Query was started for the same O_KEY used in step 1. The ACID Query blocked and did not see any uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was resumed, and COMMITTED.
4. The ACID Query completed. It returned the data as committed by the ACID Transaction.

4.3.2 Isolation Test 2 - Read-Write Conflict with Rollback

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is rolled back.

The following steps were performed to satisfy the test of isolation for a read-only and a rolled back read-write transaction:

1. An ACID Transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID Transaction was suspended prior to ROLLBACK.
2. An ACID Query was started for the same O_KEY used in step 1. The ACID Query did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was ROLLED BACK.
4. The ACID Query completed.

4.3.3 Isolation Test 3 - Write-Write Conflict with Commit

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is committed.

The following steps were performed to verify isolation of two update transactions:

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction T1 was suspended prior to COMMIT.

2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to COMMIT and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE + (DELTA1*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))

4.3.4 Isolation Test 4 - Write-Write Conflict with Rollback

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is rolled back.

The following steps were performed to verify isolation of two update transactions after the first one is rolled back:

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID transaction T1 was suspended prior to ROLLBACK.
2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to ROLLBACK and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE.

4.3.5 Isolation Test 5 – Concurrent Read and Write Transactions on Different Tables

Demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently.

The following steps were performed to demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently:

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to COMMIT.
2. Another ACID transaction, T2 was started using random values for PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEy are equal are returned.
3. ACID Transaction T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that the appropriate rows in the ORDER, LINEITEM, and HISTORY tables have been changed.

4.3.6 Isolation Test 6 – Update Transactions during Continuous Read-Only Query Stream

Demonstrate the continuous submission of arbitrary (read-only) queries against one or more tables of the database does not indefinitely delay update transactions affecting those tables from making progress.

The following steps were performed to demonstrate that the continuous submission of arbitrary (read-only) queries against one or more tables of the database:

1. A Transaction, T1, was started which executed Q21 against the qualification database, was started using a randomly selected DELTA.
2. An ACID Transaction, T2, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. T2 completed and appropriate rows in the ORDERS, LINEITEM and HISTORY tables had been changed.
4. Transaction T1 completed executing Q21.

4.4 Durability Requirements

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.2.

4.4.1 Permanent Unrecoverable Failure of Any Durable Medium

Guarantee the database and committed updates are preserved across a permanent irrecoverable failure of any single durable medium containing TPC-H database tables or recovery log tables.

During the test one disk from one of the hp StorageWorks MSA1000 containing the data files was removed; one disk from one of the hp StorageWorks MSA1000 containing the data and redo log files was removed. As all the data and redo log files resided on RAID0+1 volumes test continued uninterrupted.

4.4.2 System Crash

Guarantee the database and committed updates are preserved across an instantaneous interruption (system crash/system hang) in processing which requires the system to reboot to recover.

The system crash and memory failure tests were combined. Power to the servers was turned off during the test. The power was restored and database was restarted. The durability success file and the HISTORY table were compared and the counts matched.

4.4.3 Memory Failure

Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents).

The system crash and memory failure tests were combined. Power to the servers was turned off during the test. The power was restored and database was restarted. The durability success file and the HISTORY table were compared and the counts matched.

5.0 Clause 4: Scaling and Database Population

5.1 Initial Cardinality of Tables

The cardinality (e.g., the number of rows) of each table of the test database, as it existed at the completion of the database load (see clause 4.2.5) must be disclosed.

Table 5.1 lists the TPC Benchmark H defined tables and the row count for each table as they existed upon completion of the build.

Table 5. 1: Initial Number of Rows

Table Name	Row Count
Region	5
Nation	25
Supplier	30,000,000
Customer	450,000,000
Part	600,000,000
Partsupp	2,400,000,000
Orders	4,500,000,000
Lineitem	18,000,048,306

5.2 Distribution of Tables and Logs Across Media

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

The hp ProLiant DL740 Cluster 32P is an eight node cluster of hp ProLiant DL740 servers, depicted in Figure 1.1, consisted of:

Processors (per node) : 4 x 3.0GHz Intel Xeon Processor MP w/ 4MB cache
Memory (per node) : 8 GB
OS Disk Drives (per node) : 2 x 36GB 15krpm HDD Ultra320
NICs (per node) : 2 x on-board (used as cluster interconnects)
1 x hp NC7770 PCI-x Gigabit server adapter
Disk Controllers (per node) : 3 x hp StorageWorks fca 2214DC, 2 x hp StorageWorks fca 2214
Storage Area Network : 8 x hp StorageWorks SAN Switch 2/16
64 x hp StorageWorks MSA1000
896 x 36GB 15krpm HDD Ultra320
Total Storage : 32256GB
Cluster Interconnect : 2 x hp ProCurve Switch 4148gl

The storage area network (SAN) consisted of eight hp SAN Switch 2/16s and 64 hp StorageWorks MSA1000s. The hp ProLiant DL740 servers had three hp StorageWorks fca 2214DC (dual port) HBAs and two StorageWorks fca 2214 (single port) HBAs; connected to hp SAN Switch 2/16s. Each hp SAN Switch 2/16s had eight hp StorageWorks MSA1000s connected to it. Each hp StorageWorks MSA1000 had one RAID0+1 volume. The array accelerator cache was set to 100% read. All tables except NATION and REGION had 64 horizontal partitions; each of them resided on one of the 64 hp StorageWorks

MSA1000s. The temp files and flatfiles were also distributed across the 64 hp StorageWorks MSA1000s. Eight of the hp StorageWorks MSA1000s had two additional partitions for database redo log files. A detailed description of distribution of database files can be found in Table 5.2. The cluster interconnects were two to hp ProCurve Switch 4148gls, connected to the on-board NICs.

Table 5.2: SAN configuration and Database Layout

SAN Switch	SAN Array	Partition 1	Partition 2	Partitions3	Partition 4	Partition 5	Partition 6
1	MSA 1	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
1	MSA 2	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
1	MSA 3	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	sys	
1	MSA 4	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
1	MSA 5	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
1	MSA 6	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
1	MSA 7	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
1	MSA 8	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
2	MSA 9	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
2	MSA 10	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
2	MSA 11	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	sys	
2	MSA 12	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
2	MSA 13	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
2	MSA 14	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
2	MSA 15	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
2	MSA 16	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
3	MSA 17	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
3	MSA 18	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
3	MSA 19	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	sys	
3	MSA 20	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
3	MSA 21	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
3	MSA 22	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
3	MSA 23	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
3	MSA 24	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
4	MSA 25	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
4	MSA 26	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
4	MSA 27	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	sysaux	
4	MSA 28	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
4	MSA 29	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
4	MSA 30	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
4	MSA 31	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
4	MSA 32	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
5	MSA 33	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
5	MSA 34	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	ocr	
5	MSA 35	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
5	MSA 36	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
5	MSA 37	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
5	MSA 38	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
5	MSA 39	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
5	MSA 40	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
6	MSA 41	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
6	MSA 42	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
6	MSA 43	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	quorum	
6	MSA 44	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	perf	
6	MSA 45	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
6	MSA 46	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
6	MSA 47	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
6	MSA 48	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
7	MSA 49	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
7	MSA 50	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
7	MSA 51	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	control	
7	MSA 52	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	control	
7	MSA 53	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
7	MSA 54	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
7	MSA 55	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
7	MSA 56	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
8	MSA 57	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	redo log	redo log
8	MSA 58	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	undo	
8	MSA 59	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles	default	
8	MSA 60	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
8	MSA 61	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
8	MSA 62	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
8	MSA 63	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		
8	MSA 64	lineitem, order	customer, part, supplier, partsupp	temp	flatfiles		

5.3 Mapping of Database Partitions/Replications

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

Horizontal partitioning was used for all tables except NATION and REGION. Sections 5.2 describes the distribution of tables and redo log files. The database was not replicated.

5.4 Implementation of RAID

Implementations may use some form of RAID to ensure high availability. If used for data, auxiliary storage (e.g. indexes) or temporary space, the level of RAID used must be disclosed for each device.

RAID 0+1 was used for the entire database and redo log files.

5.5 DBGEN Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN (see Clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The supplied DBGEN version 1.3.0 was used to generate the database population for this benchmark without any modification.

5.6 Database Load time

The database load time for the test database (see clause 4.3) must be disclosed.

The database load time was 18 hours 52 minutes 0 seconds.

5.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up.

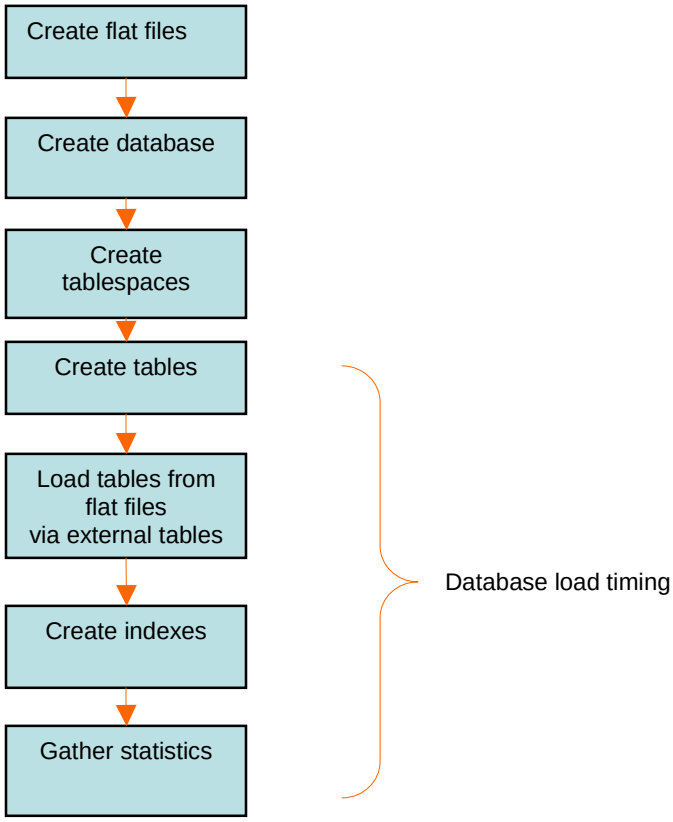
Disk Type	Number of Disks	Total Disk Space	Data Storage Ratio
36GB15krpm HDD Ultra320	896	32256GB	10.75

5.8 Database Load Mechanism Details and Illustration

The details of the database load must be disclosed, including a block diagram illustrating the overall process. Disclosure of the load procedure includes all steps, scripts, input and configuration files required to completely reproduce the test and qualification databases.

Flat files for each of the tables were created using DBGEN resided on the SAN.

Figure 5.8: Block Diagram of Database Load Process



5.9 Qualification Database Configuration

Any differences between the configuration of the qualification database and the test database must be disclosed.

The qualification database used identical scripts to create and load the data with changes to adjust for the database scale factor.

6.0 Clause 5: Performance Metrics and Execution Rules Related Items

6.1 Steps in the Power Test

The details of the steps followed to implement the power test (e.g., system boot, database restart, etc.) must be disclosed.

The following steps were used to implement the power test:

1. The system was rebooted
2. RF1 Refresh Transaction
3. Stream 00 Execution
4. RF2 Refresh Transaction.

6.2 Timing Intervals for Each Query and Refresh Function

The timing intervals (see Clause 5.3.6) for each query of the measured set and for both refresh functions must be reported for the power test.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the timing intervals for queries and refresh functions.

6.3 Number of Streams for The Throughput Test

The number of execution streams used for the throughput test must be disclosed.

Eight streams were used for the Throughput Test.

6.4 Start and End Date/Times for Each Query Stream

The start time and finish time for each query execution stream must be reported for the throughput test.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the start and stop times for the query execution streams.

6.5 Total Elapsed Time for the Measurement Interval

The total elapsed time of the measurement interval (see Clause 5.3.5) must be reported for the throughput test.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the elapsed time for the measurement interval.

6.6 Refresh Function Start Date/Time and Finish Date/Time

Start and finish time for each update function in the update stream must be reported for the throughput test.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the start and finish time for the refresh functions.

6.7 Timing Intervals for Each Query and Each Refresh Function for Each Stream

The timing intervals (see Clause 5.3.6) for each query of each stream and for each update function must be reported for the throughput test.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the timing intervals for queries and refresh functions.

6.8 Performance Metrics

The computed performance metrics, related numerical quantities and the price performance metric must be reported.

Numerical Quantities Summary section of the executive summary, which can be found in the beginning of this document, contains the performance metrics, related numerical quantities and the price performance metric.

6.9 The Performance Metric and Numerical Quantities from Both Runs

A description of the method used to determine the reproducibility of the measurement results must be reported. This must include the performance metrics (QppH and QthH) from the reproducibility runs.

Performance results from the first two executions of the TPC-H benchmark indicated the following difference for the metric points:

Run	QppH@3TB	QthH@3TB	QphH@3TB
Run 1	30430.7	16470.8	22387.9
Run 2	30958.3	16214.5	22404.8

6.11 System Activity Between Tests

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be disclosed.

The database was restarted between the two runs.
The system was restarted between runs.

7.0 Clause 6: SUT and Driver Implementation Related Items

7.1 Driver

A detailed description of how the driver performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the driver.

A single script performs all stream executions. QGEN is used to produce query text. For each power-test run:
The SQL for RF1 is submitted to the database
Then the queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4
The SQL for RF2 is submitted to the database.

7.2 Implementation Specific Layer (ISL)

If an implementation-specific layer is used, then a detailed description of how it performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the implementation-specific layer.

The source code for the qexec utility can be found in Appendix F.

7.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2.9 is used, such used must be disclosed.

Profile-directed optimization was not used.

8.0 Clause 7: Pricing Related Items

8.1 Hardware and Software Used

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 date. 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.

A detailed list of hardware and software used in the priced system is included in the pricing sheet in the executive summary. All prices are currently effective. Third-party price quotations are included in Appendix G.

8.2 Total 3 Year Price

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

A detailed pricing sheet of all the hardware and software used in this configuration and the 3-year maintenance costs, demonstrating the computation of the total 3-year price of the configuration, is included in the executive summary at the beginning of this document. For a large purchase and cash discount, this purchase qualifies for a 17% discount from Hewlett Packard Company.

8.3 Availability Date

The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the availability date reported on the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided.

All hardware and software are available at the time of publication.

8.4 Country-Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country-specific priced configuration. Country-specific pricing is subject to Clause 7.1.7.

The configuration is priced for the United States of America.

9.0 Clause 9: Related Items

9.1 Auditors' Report

The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.

Lorna Livingtree of Performance Metrics Inc audited this implementation of the TPC Benchmark H.

Performance Metrics Inc.
137 Yankton St., Suite 101
Folsom, CA 95630
USA

Email: lorna@perfmetrics.com
Phone: (916) 985-1131

TPC Benchmark H Full Disclosure Report and other information can be downloaded from Transaction Processing Performance Council web site at www.tpc.org.

PERFORMANCE METRICS INC.
TPC Certified Auditors

February 27, 2004

Mr. Raghunath Othayoth
Mr. Bryon Georgson
Database Performance Engineers
HP Database Performance Lab
20555 SH 249
Houston, TX 77070

I have verified the TPC Benchmark™ H for the following configuration:

Platform: HP Proliant DL740 Cluster 32P
Database Manager: Oracle Database 10g
Operating System: Red Hat Enterprise Linux AS 3

Each node of the cluster contained:

CPU's	Memory	Total Disks	QppH@100GB	QthH@100GB	QphH@100GB
4 Intel Xeon MP @ 2.8 Ghz	8 GB	2 OS disk 36GB, 15Krpm	**30,430.7**	**16,470.8**	**22,387.9**

The entire cluster was supported by a storage area network of 896 @ 36 GB, 15Krpm.

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 database tables were defined with the proper columns, layout and sizes.
- The tested database was correctly scaled and populated for 3000GB using DBGEN.
- The qualification database layout was identical to the tested database except for the size of the files.
- The query text was verified to use only compliant variants and minor modifications.
- The executable query text was generated by QGEN and submitted through an interactive interface verified to be compliant.

- The validation of the query text against the qualification database produced compliant results.
- The refresh functions were properly implemented and executed the correct number of inserts and deletes.
- The load timing was properly measured and reported.
- The execution times were correctly measured and reported.
- The performance metrics were correctly computed and reported.
- The repeatability of the measurement was verified.
- The ACID properties were tested and verified.
- Sufficient mirrored log space was present on the tested system.
- The system pricing was checked for major components and maintenance.
- The price quotes were checked and found to be compliant.
- The executive summary pages of the FDR were verified for accuracy.

Auditor's Notes:
None.

Sincerely,

Lorna Livingtree
Auditor

Appendix A: Tunable Parameters

audit_build.ora

```
aq_tm_processes                 = 0
audit_trail                     = FALSE
compatible                      = 10.0.0.0
control_files                   =
(/home/oracle/dev/raw/control_001,
/home/oracle/dev/raw/control_002)
cpu_count                     = 4
db_block_checksum               = false
db_block_size                   = 16384
db_cache_size                   = 200m
db_file_multiblock_read_count   = 128
db_files                        = 500
db_name                         = 10i
db_writer_processes             = 4
dml_locks                       = 40000
enqueue_resources               = 40000
global_names                    = FALSE
instance_name                   = inst_1
shared_pool_size                = 512m
large_pool_size                 = 300m
log_buffer                      = 4194304
log_checkpoints_to_alert        = true
max_dump_file_size              = unlimited
nls_date_format                 = YYYY-MM-DD
open_cursors                    = 600
optimizer_mode                  = CHOOSE
optimizer_features_enable       = 10.0.0.1
parallel_adaptive_multi_user    = TRUE
parallel_execution_message_size = 16384
parallel_max_servers            = 140
parallel_min_servers            = 140
pga_aggregate_target            = 6g
processes                       = 1000
query_rewrite_enabled           = true
recovery_parallelism            = 8
replication_dependency_tracking = false
statistics_level                = BASIC
transactions                    = 10
undo_management                 = auto
undo_retention                      = 500000
```

audit_run.ora

```
aq_tm_processes                 = 0
audit_trail                     = FALSE
compatible                      = 10.0.0.0
control_files                   =
(/home/oracle/dev/raw/control_001,
/home/oracle/dev/raw/control_002)
cpu_count                     = 4
db_block_checksum               = false
db_block_size                   = 16384
db_cache_size                   = 1250m
db_file_multiblock_read_count   = 128
db_files                        = 500
db_name                         = 10i
db_writer_processes             = 4
dml_locks                       = 40000
enqueue_resources               = 40000
global_names                    = FALSE
instance_name                   = inst_1
shared_pool_size                = 512m
large_pool_size                 = 300m
log_buffer                      = 4194304
log_checkpoints_to_alert        = true
max_dump_file_size              = unlimited
nls_date_format                 = YYYY-MM-DD
open_cursors                    = 600
optimizer_mode                  = CHOOSE
optimizer_features_enable       = 10.0.0.1
parallel_adaptive_multi_user    = TRUE
parallel_execution_message_size = 16384
parallel_max_servers            = 140
parallel_min_servers            = 140
pga_aggregate_target            = 6g
processes                       = 1000
query_rewrite_enabled           = true
recovery_parallelism            = 8
replication_dependency_tracking = false
statistics_level                = BASIC
transactions                    = 10
undo_retention                      = 500000
```

init_h1.ora

```
instance_number =1
thread  =1
undo_management                 = auto
UNDO_TABLESPACE                 = ts_undo1
cluster_database                = true
cluster_interconnects           = 10.1.0.1
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h2.ora

```
instance_number =2
thread  =2
undo_management                 = auto
UNDO_TABLESPACE                 = ts_undo2
cluster_database                = true
cluster_interconnects           = 10.1.0.2
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h3.ora

```
instance_number =3
thread  =3
undo_management                 = auto
UNDO_TABLESPACE                 = ts_undo3
cluster_database                = true
cluster_interconnects           = 10.1.0.3
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h4.ora

```
instance_number =4
thread  =4
```

```
undo_management                  = auto
UNDO_TABLESPACE                  = ts_undo4
cluster_database                 = true
cluster_interconnects            = 10.1.0.4
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h5.ora

```
instance_number =5
thread  =5
undo_management                  = auto
UNDO_TABLESPACE                  = ts_undo5
cluster_database                 = true
cluster_interconnects            = 10.1.0.5
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h6.ora

```
instance_number =6
thread  =6
undo_management                  = auto
UNDO_TABLESPACE                  = ts_undo6
cluster_database                 = true
cluster_interconnects            = 10.1.0.6
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h7.ora

```
instance_number =7
thread  =7
undo_management                  = auto
UNDO_TABLESPACE                  = ts_undo7
cluster_database                 = true
cluster_interconnects            = 10.1.0.7
ifile=/home/oracle/tpch/dbs/init_run.ora
```

init_h8.ora

```
instance_number =8
thread  =8
undo_management                  = auto
UNDO_TABLESPACE                  = ts_undo8
cluster_database                 = true
cluster_interconnects            = 10.1.0.8
ifile=/home/oracle/tpch/dbs/init_run.ora
```

rc.local

```
#!/bin/sh
#
# This script will be executed *after* all the
other init scripts.
# You can put your own initialization stuff in
here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local

rmmod qla2300
# defaults queue=64 timer=3
# Andy suggests q=32 t=0
insmod qla2300 ql2xmaxqdepth=32
ql2xintrdelaytimer=0
sh /root/mkraw.sh

rdate -s dbgate

#echo  6442450944 > /proc/sys/kernel/shmmax
echo  3000000000 > /proc/sys/kernel/shmmax
echo  1048576 > /proc/sys/fs/aio-max-nr


mount -o ro /dev/sdb6 /home/oracle/dev/ff_1
mount -o ro /dev/sdl6 /home/oracle/dev/ff_2
mount -o ro /dev/sdv6 /home/oracle/dev/ff_3
mount -o ro /dev/sdaf6 /home/oracle/dev/ff_4
mount -o ro /dev/sdap6 /home/oracle/dev/ff_5
mount -o ro /dev/sdaz6 /home/oracle/dev/ff_6
mount -o ro /dev/sdbj6 /home/oracle/dev/ff_7
mount -o ro /dev/sdbt6 /home/oracle/dev/ff_8
mount -o ro /dev/sdd6 /home/oracle/dev/ff_9
mount -o ro /dev/sdn6 /home/oracle/dev/ff_10
mount -o ro /dev/sdx6 /home/oracle/dev/ff_11
mount -o ro /dev/sdah6 /home/oracle/dev/ff_12
mount -o ro /dev/sdar6 /home/oracle/dev/ff_13
mount -o ro /dev/sdbb6 /home/oracle/dev/ff_14
mount -o ro /dev/sdbl6 /home/oracle/dev/ff_15
mount -o ro /dev/sdbv6 /home/oracle/dev/ff_16
mount -o ro /dev/sde6 /home/oracle/dev/ff_17
mount -o ro /dev/sdo6 /home/oracle/dev/ff_18
mount -o ro /dev/sdy6 /home/oracle/dev/ff_19
mount -o ro /dev/sdai6 /home/oracle/dev/ff_20
mount -o ro /dev/sdas6 /home/oracle/dev/ff_21
mount -o ro /dev/sdbc6 /home/oracle/dev/ff_22
mount -o ro /dev/sdbm6 /home/oracle/dev/ff_23
mount -o ro /dev/sdbw6 /home/oracle/dev/ff_24
mount -o ro /dev/sdf6 /home/oracle/dev/ff_25
mount -o ro /dev/sdp6 /home/oracle/dev/ff_26
mount -o ro /dev/sdz6 /home/oracle/dev/ff_27
mount -o ro /dev/sdaj6 /home/oracle/dev/ff_28
mount -o ro /dev/sdat6 /home/oracle/dev/ff_29
mount -o ro /dev/sdbd6 /home/oracle/dev/ff_30
mount -o ro /dev/sdbn6 /home/oracle/dev/ff_31
mount -o ro /dev/sdbx6 /home/oracle/dev/ff_32
mount -o ro /dev/sdg6 /home/oracle/dev/ff_33
mount -o ro /dev/sdq6 /home/oracle/dev/ff_34
mount -o ro /dev/sdaa6 /home/oracle/dev/ff_35
mount -o ro /dev/sdak6 /home/oracle/dev/ff_36
mount -o ro /dev/sdau6 /home/oracle/dev/ff_37
mount -o ro /dev/sdbe6 /home/oracle/dev/ff_38
mount -o ro /dev/sdbo6 /home/oracle/dev/ff_39
mount -o ro /dev/sdby6 /home/oracle/dev/ff_40
mount -o ro /dev/sdh6 /home/oracle/dev/ff_41
mount -o ro /dev/sdr6 /home/oracle/dev/ff_42
mount -o ro /dev/sdab6 /home/oracle/dev/ff_43
mount -o ro /dev/sdal6 /home/oracle/dev/ff_44
mount -o ro /dev/sdav6 /home/oracle/dev/ff_45
mount -o ro /dev/sdbf6 /home/oracle/dev/ff_46
mount -o ro /dev/sdbp6 /home/oracle/dev/ff_47
mount -o ro /dev/sdbz6 /home/oracle/dev/ff_48
mount -o ro /dev/sdi6 /home/oracle/dev/ff_49
mount -o ro /dev/sds6 /home/oracle/dev/ff_50
mount -o ro /dev/sdac6 /home/oracle/dev/ff_51
mount -o ro /dev/sdam6 /home/oracle/dev/ff_52
mount -o ro /dev/sdaw6 /home/oracle/dev/ff_53
mount -o ro /dev/sdbg6 /home/oracle/dev/ff_54
mount -o ro /dev/sdbq6 /home/oracle/dev/ff_55
mount -o ro /dev/sdca6 /home/oracle/dev/ff_56
mount -o ro /dev/sdj6 /home/oracle/dev/ff_57
mount -o ro /dev/sdt6 /home/oracle/dev/ff_58
mount -o ro /dev/sdad6 /home/oracle/dev/ff_59
mount -o ro /dev/sdan6 /home/oracle/dev/ff_60
```

```
mount -o ro /dev/sdax6 /home/oracle/dev/ff_61
mount -o ro /dev/sdbh6 /home/oracle/dev/ff_62
mount -o ro /dev/sdbr6 /home/oracle/dev/ff_63
mount -o ro /dev/sdcb6 /home/oracle/dev/ff_64

#/etc/init.d/netdump start
#ifdown eth0
#ifdown eth1
#ifdown eth2
#rmmod tg3
#insmod /root/bcm5700.o
#ifup eth0
#ifup eth2
#insmod /lib/modules/2.4.21-
1.1931.2.421.entcustom/kernel/drivers/net/e1000/
e1000.o

/sbin/ifconfig eth0 txqueuelen 10000
/sbin/ifconfig eth1 txqueuelen 10000
/sbin/ifconfig eth2 txqueuelen 10000


echo 10 > /proc/irq/20/smp_affinity
echo 20 > /proc/irq/21/smp_affinity
echo 40 > /proc/irq/22/smp_affinity
echo 80 > /proc/irq/23/smp_affinity
echo 10 > /proc/irq/27/smp_affinity
echo 20 > /proc/irq/29/smp_affinity
echo 40 > /proc/irq/30/smp_affinity
echo 80 > /proc/irq/31/smp_affinity
echo 10 > /proc/irq/44/smp_affinity
echo 20 > /proc/irq/43/smp_affinity
echo 40 > /proc/irq/25/smp_affinity


#ifconfig eth2 mtu 9000
echo  3000000000 > /proc/sys/kernel/shmmax
```

sysctl.conf

```
# Kernel sysctl configuration file for Red Hat
Linux
#
# For binary values, 0 is disabled, 1 is
enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Controls the System Request debugging
functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the
PID to the core filename.
# Useful for debugging multi-threaded
applications.
kernel.core_uses_pid = 1

#net.core.rmem_max = 2097152
#net.core.rmem_default = 2097152
#net.core.wmem_max = 2097152
#net.core.wmem_default = 2097152

net.core.rmem_max = 524288
net.core.rmem_default = 524288
net.core.wmem_max = 524288
net.core.wmem_default = 524288

fs.aio-max-size = 2097152
kernel.shmmax = 3000000000
```

Appendix B: Database Build Scripts

_____
3TB.dat
_____

```
############################################################
#############
# preprocessing-like directives

%b-preproc

*sql
\echo "{}" > script*getenv(BUMPX_CTR).sql
\sqlplus /NOLOG <<!
\set echo on;
\set timing on;
\set termout on;
\connect / as sysdba;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now from dual;
\@script*getenv(BUMPX_CTR).sql;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now from dual;
\exit;
\!
\/bin/rm script*getenv(BUMPX_CTR).sql;


*load1
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=1
############################################################
##################
# Database Creation Phase
*sql
{
shutdown abort;
}
*wait
# creating database  and undo tablespace
*sql
{
startup pfile= /home/oracle/tpch/dbs/init_run.ora nomount;
create database
  controlfile reuse
  logfile '/home/oracle/dev/raw/log_1_1' size 4096m reuse,
          '/home/oracle/dev/raw/log_1_2' size 4096m reuse
  datafile '/home/oracle/dev/raw/sys_1' size 1024m reuse
  sysaux datafile '/home/oracle/dev/raw/sysaux_1' size 512m reuse
  undo tablespace ts_undo1
      datafile  '/home/oracle/dev/raw/undo_1' size 38150m reuse
  maxdatafiles 4000
  maxinstances 8
;
}
*wait
*sql
{
create  undo tablespace ts_undo2
  datafile '/home/oracle/dev/raw/undo_2' size 38150m reuse
;
}
*sql
{
create  undo tablespace ts_undo3
  datafile '/home/oracle/dev/raw/undo_3' size 38150m reuse
;
}
*sql
{
create  undo tablespace ts_undo4
  datafile '/home/oracle/dev/raw/undo_4' size 38150m reuse
;
}
*sql
{
create  undo tablespace ts_undo5
  datafile '/home/oracle/dev/raw/undo_5' size 38150m reuse
;
}
*sql
{
create  undo tablespace ts_undo6
  datafile '/home/oracle/dev/raw/undo_6' size 38150m reuse
;
}
*sql
{
create  undo tablespace ts_undo7
  datafile '/home/oracle/dev/raw/undo_7' size 38150m reuse
;
}
*sql
{
create  undo tablespace ts_undo8
  datafile '/home/oracle/dev/raw/undo_8' size 38150m reuse
;
}
# creating extra logfile threads for rac 8 nodes
*sql
{
alter database add logfile thread 2
  '/home/oracle/dev/raw/log_2_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_2_2' size 4096m reuse;
alter database enable public thread 2;
}
*sql
{
alter database add logfile thread 3
  '/home/oracle/dev/raw/log_3_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_3_2' size 4096m reuse;
alter database enable public thread 3;
}
*sql
{
alter database add logfile thread 4
  '/home/oracle/dev/raw/log_4_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_4_2' size 4096m reuse;
```

```
alter database enable public thread 4;
}
*sql
{
alter database add logfile thread 5
  '/home/oracle/dev/raw/log_5_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_5_2' size 4096m reuse;
alter database enable public thread 5;
}
*sql
{
alter database add logfile thread 6
  '/home/oracle/dev/raw/log_6_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_6_2' size 4096m reuse;
alter database enable public thread 6;
}
*sql
{
alter database add logfile thread 7
  '/home/oracle/dev/raw/log_7_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_7_2' size 4096m reuse;
alter database enable public thread 7;
}
*sql
{
alter database add logfile thread 8
  '/home/oracle/dev/raw/log_8_1' size 4096m reuse,
  '/home/oracle/dev/raw/log_8_2' size 4096m reuse;
alter database enable public thread 8;
}
# building data dictionary
*sql
{
set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@?/sqlplus/admin/pupbld.sql;
spool off
}
*wait
*bgoff
%e-dbcre
%b-sctso
*bgon=300
############################################################
##################
# Schema Creation Phase - datafiles only (no tables or users)
# creating data tablespaces, datafiles
# creating tpch's ts_one tablespace

*sql
{
--drop tablespace ts_default including contents;
create tablespace ts_default
datafile '/home/oracle/dev/raw/default_1' size 1024m reuse
extent management local
autoallocate
;
}
*sql
{
-- drop tablespace ts_lo1
create tablespace ts_lo1
datafile '/home/oracle/dev/raw/lo_1' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo2
create tablespace ts_lo2
datafile '/home/oracle/dev/raw/lo_2' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo3
create tablespace ts_lo3
datafile '/home/oracle/dev/raw/lo_3' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo4
create tablespace ts_lo4
datafile '/home/oracle/dev/raw/lo_4' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo5
create tablespace ts_lo5
datafile '/home/oracle/dev/raw/lo_5' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo6
create tablespace ts_lo6
datafile '/home/oracle/dev/raw/lo_6' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo7
create tablespace ts_lo7
datafile '/home/oracle/dev/raw/lo_7' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo8
create tablespace ts_lo8
datafile '/home/oracle/dev/raw/lo_8' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo9
```

```
create tablespace ts_lo9
datafile '/home/oracle/dev/raw/lo_9' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo10
create tablespace ts_lo10
datafile '/home/oracle/dev/raw/lo_10' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo11
create tablespace ts_lo11
datafile '/home/oracle/dev/raw/lo_11' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo12
create tablespace ts_lo12
datafile '/home/oracle/dev/raw/lo_12' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo13
create tablespace ts_lo13
datafile '/home/oracle/dev/raw/lo_13' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo14
create tablespace ts_lo14
datafile '/home/oracle/dev/raw/lo_14' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo15
create tablespace ts_lo15
datafile '/home/oracle/dev/raw/lo_15' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo16
create tablespace ts_lo16
datafile '/home/oracle/dev/raw/lo_16' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo17
create tablespace ts_lo17
datafile '/home/oracle/dev/raw/lo_17' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo18
create tablespace ts_lo18
datafile '/home/oracle/dev/raw/lo_18' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo19
create tablespace ts_lo19
datafile '/home/oracle/dev/raw/lo_19' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo20
create tablespace ts_lo20
datafile '/home/oracle/dev/raw/lo_20' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo21
create tablespace ts_lo21
datafile '/home/oracle/dev/raw/lo_21' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo22
create tablespace ts_lo22
datafile '/home/oracle/dev/raw/lo_22' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo23
create tablespace ts_lo23
datafile '/home/oracle/dev/raw/lo_23' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo24
create tablespace ts_lo24
datafile '/home/oracle/dev/raw/lo_24' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
-- drop tablespace ts_lo25
create tablespace ts_lo25
datafile '/home/oracle/dev/raw/lo_25' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo26
create tablespace ts_lo26
datafile '/home/oracle/dev/raw/lo_26' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo27
create tablespace ts_lo27
datafile '/home/oracle/dev/raw/lo_27' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo28
create tablespace ts_lo28
datafile '/home/oracle/dev/raw/lo_28' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo29
create tablespace ts_lo29
datafile '/home/oracle/dev/raw/lo_29' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo30
create tablespace ts_lo30
datafile '/home/oracle/dev/raw/lo_30' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo31
create tablespace ts_lo31
datafile '/home/oracle/dev/raw/lo_31' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo32
create tablespace ts_lo32
datafile '/home/oracle/dev/raw/lo_32' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_l33
create tablespace ts_lo33
datafile '/home/oracle/dev/raw/lo_33' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo34
create tablespace ts_lo34
datafile '/home/oracle/dev/raw/lo_34' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo35
create tablespace ts_lo35
datafile '/home/oracle/dev/raw/lo_35' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo36
create tablespace ts_lo36
datafile '/home/oracle/dev/raw/lo_36' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo37
create tablespace ts_lo37
datafile '/home/oracle/dev/raw/lo_37' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo38
create tablespace ts_lo38
datafile '/home/oracle/dev/raw/lo_38' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo39
create tablespace ts_lo39
datafile '/home/oracle/dev/raw/lo_39' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo40
```

```
create tablespace ts_lo40
datafile '/home/oracle/dev/raw/lo_40' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo41
create tablespace ts_lo41
datafile '/home/oracle/dev/raw/lo_41' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo42
create tablespace ts_lo42
datafile '/home/oracle/dev/raw/lo_42' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo43
create tablespace ts_lo43
datafile '/home/oracle/dev/raw/lo_43' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo44
create tablespace ts_lo44
datafile '/home/oracle/dev/raw/lo_44' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo45
create tablespace ts_lo45
datafile '/home/oracle/dev/raw/lo_45' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo46
create tablespace ts_lo46
datafile '/home/oracle/dev/raw/lo_46' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo47
create tablespace ts_lo47
datafile '/home/oracle/dev/raw/lo_47' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo48
create tablespace ts_lo48
datafile '/home/oracle/dev/raw/lo_48' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo1
create tablespace ts_lo49
datafile '/home/oracle/dev/raw/lo_49' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_l50
create tablespace ts_lo50
datafile '/home/oracle/dev/raw/lo_50' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo51
create tablespace ts_lo51
datafile '/home/oracle/dev/raw/lo_51' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo52
create tablespace ts_lo52
datafile '/home/oracle/dev/raw/lo_52' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo53
create tablespace ts_lo53
datafile '/home/oracle/dev/raw/lo_53' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo54
create tablespace ts_lo54
datafile '/home/oracle/dev/raw/lo_54' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo55
create tablespace ts_lo55
datafile '/home/oracle/dev/raw/lo_55' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
-- drop tablespace ts_lo56
create tablespace ts_lo56
datafile '/home/oracle/dev/raw/lo_56' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo57
create tablespace ts_lo57
datafile '/home/oracle/dev/raw/lo_57' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo58
create tablespace ts_lo58
datafile '/home/oracle/dev/raw/lo_58' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo59
create tablespace ts_lo59
datafile '/home/oracle/dev/raw/lo_59' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo60
create tablespace ts_lo60
datafile '/home/oracle/dev/raw/lo_60' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo61
create tablespace ts_lo61
datafile '/home/oracle/dev/raw/lo_61' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo62
create tablespace ts_lo62
datafile '/home/oracle/dev/raw/lo_62' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo63
create tablespace ts_lo63
datafile '/home/oracle/dev/raw/lo_63' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_lo64
create tablespace ts_lo64
datafile '/home/oracle/dev/raw/lo_64' size 59999m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest1
create tablespace ts_rest1
datafile '/home/oracle/dev/raw/r_1' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest2
create tablespace ts_rest2
datafile '/home/oracle/dev/raw/r_2' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest3
create tablespace ts_rest3
datafile '/home/oracle/dev/raw/r_3' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest4
create tablespace ts_rest4
datafile '/home/oracle/dev/raw/r_4' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest5
create tablespace ts_rest5
datafile '/home/oracle/dev/raw/r_5' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest6
create tablespace ts_rest6
datafile '/home/oracle/dev/raw/r_6' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest7
```

```
create tablespace ts_rest7
datafile '/home/oracle/dev/raw/r_7' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest8
create tablespace ts_rest8
datafile '/home/oracle/dev/raw/r_8' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest9
create tablespace ts_rest9
datafile '/home/oracle/dev/raw/r_9' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest10
create tablespace ts_rest10
datafile '/home/oracle/dev/raw/r_10' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest11
create tablespace ts_rest11
datafile '/home/oracle/dev/raw/r_11' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest12
create tablespace ts_rest12
datafile '/home/oracle/dev/raw/r_12' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest13
create tablespace ts_rest13
datafile '/home/oracle/dev/raw/r_13' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest14
create tablespace ts_rest14
datafile '/home/oracle/dev/raw/r_14' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest15
create tablespace ts_rest15
datafile '/home/oracle/dev/raw/r_15' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest16
create tablespace ts_rest16
datafile '/home/oracle/dev/raw/r_16' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest17
create tablespace ts_rest17
datafile '/home/oracle/dev/raw/r_17' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest18
create tablespace ts_rest18
datafile '/home/oracle/dev/raw/r_18' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest19
create tablespace ts_rest19
datafile '/home/oracle/dev/raw/r_19' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest20
create tablespace ts_rest20
datafile '/home/oracle/dev/raw/r_20' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest21
create tablespace ts_rest21
datafile '/home/oracle/dev/raw/r_21' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest22
create tablespace ts_rest22
datafile '/home/oracle/dev/raw/r_22' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
-- drop tablespace ts_rest23
create tablespace ts_rest23
datafile '/home/oracle/dev/raw/r_23' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest24
create tablespace ts_rest24
datafile '/home/oracle/dev/raw/r_24' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest25
create tablespace ts_rest25
datafile '/home/oracle/dev/raw/r_25' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest26
create tablespace ts_rest26
datafile '/home/oracle/dev/raw/r_26' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest27
create tablespace ts_rest27
datafile '/home/oracle/dev/raw/r_27' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest28
create tablespace ts_rest28
datafile '/home/oracle/dev/raw/r_28' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest29
create tablespace ts_rest29
datafile '/home/oracle/dev/raw/r_29' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest30
create tablespace ts_rest30
datafile '/home/oracle/dev/raw/r_30' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest31
create tablespace ts_rest31
datafile '/home/oracle/dev/raw/r_31' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest32
create tablespace ts_rest32
datafile '/home/oracle/dev/raw/r_32' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_l33
create tablespace ts_rest33
datafile '/home/oracle/dev/raw/r_33' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest34
create tablespace ts_rest34
datafile '/home/oracle/dev/raw/r_34' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest35
create tablespace ts_rest35
datafile '/home/oracle/dev/raw/r_35' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest36
create tablespace ts_rest36
datafile '/home/oracle/dev/raw/r_36' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest37
create tablespace ts_rest37
datafile '/home/oracle/dev/raw/r_37' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest38
```

```
create tablespace ts_rest38
datafile '/home/oracle/dev/raw/r_38' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest39
create tablespace ts_rest39
datafile '/home/oracle/dev/raw/r_39' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest40
create tablespace ts_rest40
datafile '/home/oracle/dev/raw/r_40' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest41
create tablespace ts_rest41
datafile '/home/oracle/dev/raw/r_41' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest42
create tablespace ts_rest42
datafile '/home/oracle/dev/raw/r_42' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest43
create tablespace ts_rest43
datafile '/home/oracle/dev/raw/r_43' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest44
create tablespace ts_rest44
datafile '/home/oracle/dev/raw/r_44' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest45
create tablespace ts_rest45
datafile '/home/oracle/dev/raw/r_45' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest46
create tablespace ts_rest46
datafile '/home/oracle/dev/raw/r_46' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest47
create tablespace ts_rest47
datafile '/home/oracle/dev/raw/r_47' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest48
create tablespace ts_rest48
datafile '/home/oracle/dev/raw/r_48' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest1
create tablespace ts_rest49
datafile '/home/oracle/dev/raw/r_49' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_l50
create tablespace ts_rest50
datafile '/home/oracle/dev/raw/r_50' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest51
create tablespace ts_rest51
datafile '/home/oracle/dev/raw/r_51' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest52
create tablespace ts_rest52
datafile '/home/oracle/dev/raw/r_52' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest53
create tablespace ts_rest53
datafile '/home/oracle/dev/raw/r_53' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
-- drop tablespace ts_rest54
create tablespace ts_rest54
datafile '/home/oracle/dev/raw/r_54' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest55
create tablespace ts_rest55
datafile '/home/oracle/dev/raw/r_55' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest56
create tablespace ts_rest56
datafile '/home/oracle/dev/raw/r_56' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest57
create tablespace ts_rest57
datafile '/home/oracle/dev/raw/r_57' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest58
create tablespace ts_rest58
datafile '/home/oracle/dev/raw/r_58' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest59
create tablespace ts_rest59
datafile '/home/oracle/dev/raw/r_59' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest60
create tablespace ts_rest60
datafile '/home/oracle/dev/raw/r_60' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest61
create tablespace ts_rest61
datafile '/home/oracle/dev/raw/r_61' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest62
create tablespace ts_rest62
datafile '/home/oracle/dev/raw/r_62' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest63
create tablespace ts_rest63
datafile '/home/oracle/dev/raw/r_63' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
-- drop tablespace ts_rest64
create tablespace ts_rest64
datafile '/home/oracle/dev/raw/r_64' size 24575m reuse
extent management dictionary default storage (initial 320m next 10m
maxextents unlimited pctincrease 0)
;
}
*wait
# creating tpch's ts_temp tablespace
*sql
{
--drop tablespace ts_temp including contents;
create temporary tablespace ts_temp
tempfile '/home/oracle/dev/raw/t_1' size 38154m reuse
extent management local
uniform size 10M
;
}
*wait
#adding tpch's ts_temp add datafiles
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_2' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_3' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_4' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_5' size 38154m reuse
;
}
*sql
{
```

```
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_6' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_7' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_8' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_9' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_10' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_11' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_12' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_13' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_14' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_15' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_16' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_17' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_18' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_19' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_20' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_21' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_22' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_23' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_24' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_25' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_26' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_27' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_28' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_29' size 38154m reuse
```

```
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_30' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_31' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_32' size 38154m reuse
;
}
*wait
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_33' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_34' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_35' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_36' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_37' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_38' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_39' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_40' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_41' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_42' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_43' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_44' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_45' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_46' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_47' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_48' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_49' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_50' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_51' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_52' size 38154m reuse
;
```

```
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_53' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_54' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_55' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_56' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_57' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_58' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_59' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_60' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_61' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_62' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_63' size 38154m reuse
;
}
*sql
{
alter tablespace ts_temp
add tempfile '/home/oracle/dev/raw/t_64' size 38154m reuse
;
}
*wait
*bgoff
%e-sctso
%b-dapop
*bgon=1
############################################################
##################
# Schema Creation Phase - User and Tables
#  AND  Database Population Phase
#
# creating tpch user
#*sql
#{
#shutdown abort;
#}
#*wait
#*sql
#{
#startup pfile=INIT_ORA_PATH/init_run.ora
#}
#*wait
*sql
{
drop user tpch cascade;
grant DBA
to tpch identified by tpch;
}
*wait
*sql
{
connect tpch/tpch;
drop directory data_dir;
create directory data_dir as '/home/oracle/OraHome1/dbs/flatfiles';
}
*sql
{
connect tpch/tpch;
drop table l_et;
create table l_et(
   l_orderkey          number ,
   l_partkey           number ,
   l_suppkey           number ,
   l_linenumber        number ,
   l_quantity          number ,
   l_extendedprice     number ,
   l_discount          number ,
   l_tax               number ,
   l_returnflag        char(1) ,
   l_linestatus        char(1) ,
   l_shipdate          date ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipinstruct      char(25) ,
   l_shipmode          char(10) ,
   l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
                )
```

```
            location (
'L1', 'L2', 'L3', 'L4', 'L5', 'L6',
'L7', 'L8', 'L9', 'L10', 'L11', 'L12',
'L13', 'L14', 'L15', 'L16', 'L17', 'L18',
'L19', 'L20', 'L21', 'L22', 'L23', 'L24',
'L25', 'L26', 'L27', 'L28', 'L29', 'L30',
'L31', 'L32', 'L33', 'L34', 'L35', 'L36',
'L37', 'L38', 'L39', 'L40', 'L41', 'L42',
'L43', 'L44', 'L45', 'L46', 'L47', 'L48',
'L49', 'L50', 'L51', 'L52', 'L53', 'L54',
'L55', 'L56', 'L57', 'L58', 'L59', 'L60',
'L61', 'L62', 'L63', 'L64'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table o_et;
create table o_et(
   o_orderkey          number ,
   o_custkey           number ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_orderdate         date ,
   o_orderpriority     char(15) ,
   o_clerk             char(15) ,
   o_shippriority      number ,
   o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'O1', 'O2', 'O3', 'O4', 'O5', 'O6',
'O7', 'O8', 'O9', 'O10', 'O11', 'O12',
'O13', 'O14', 'O15', 'O16', 'O17', 'O18',
'O19', 'O20', 'O21', 'O22', 'O23', 'O24',
'O25', 'O26', 'O27', 'O28', 'O29', 'O30',
'O31', 'O32', 'O33', 'O34', 'O35', 'O36',
'O37', 'O38', 'O39', 'O40', 'O41', 'O42',
'O43', 'O44', 'O45', 'O46', 'O47', 'O48',
'O49', 'O50', 'O51', 'O52', 'O53', 'O54',
'O55', 'O56', 'O57', 'O58', 'O59', 'O60',
'O61', 'O62', 'O63', 'O64'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table ps_et;
create table ps_et(
   ps_partkey          number ,
   ps_suppkey          number ,
   ps_availqty         number ,
   ps_supplycost       number ,
   ps_comment          varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'PS1', 'PS2', 'PS3', 'PS4', 'PS5', 'PS6',
'PS7', 'PS8', 'PS9', 'PS10', 'PS11', 'PS12',
'PS13', 'PS14', 'PS15', 'PS16', 'PS17', 'PS18',
'PS19', 'PS20', 'PS21', 'PS22', 'PS23', 'PS24',
'PS25', 'PS26', 'PS27', 'PS28', 'PS29', 'PS30',
'PS31', 'PS32', 'PS33', 'PS34', 'PS35', 'PS36',
'PS37', 'PS38', 'PS39', 'PS40', 'PS41', 'PS42',
'PS43', 'PS44', 'PS45', 'PS46', 'PS47', 'PS48',
'PS49', 'PS50', 'PS51', 'PS52', 'PS53', 'PS54',
'PS55', 'PS56', 'PS57', 'PS58', 'PS59', 'PS60',
'PS61', 'PS62', 'PS63', 'PS64'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table p_et;
create table p_et(
   p_partkey           number ,
   p_name              varchar(55) ,
   p_mfgr              char(25) ,
   p_brand             char(10) ,
   p_type              varchar(25) ,
   p_size              number ,
   p_container         char(10) ,
   p_retailprice       number ,
   p_comment           varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'P1', 'P2', 'P3', 'P4', 'P5', 'P6',
'P7', 'P8', 'P9', 'P10', 'P11', 'P12',
'P13', 'P14', 'P15', 'P16', 'P17', 'P18',
'P19', 'P20', 'P21', 'P22', 'P23', 'P24',
'P25', 'P26', 'P27', 'P28', 'P29', 'P30',
'P31', 'P32', 'P33', 'P34', 'P35', 'P36',
'P37', 'P38', 'P39', 'P40', 'P41', 'P42',
'P43', 'P44', 'P45', 'P46', 'P47', 'P48',
'P49', 'P50', 'P51', 'P52', 'P53', 'P54',
'P55', 'P56', 'P57', 'P58', 'P59', 'P60',
'P61', 'P62', 'P63', 'P64'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table c_et;
create table c_et(
   c_custkey           number ,
```

```
   c_name          varchar(25) ,
   c_address       varchar(40) ,
   c_nationkey      number ,
   c_phone         char(15) ,
   c_acctbal       number ,
   c_mktsegment     char(10) ,
   c_comment       varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'C1', 'C2', 'C3', 'C4', 'C5', 'C6',
'C7', 'C8', 'C9', 'C10', 'C11', 'C12',
'C13', 'C14', 'C15', 'C16', 'C17', 'C18',
'C19', 'C20', 'C21', 'C22', 'C23', 'C24',
'C25', 'C26', 'C27', 'C28', 'C29', 'C30',
'C31', 'C32', 'C33', 'C34', 'C35', 'C36',
'C37', 'C38', 'C39', 'C40', 'C41', 'C42',
'C43', 'C44', 'C45', 'C46', 'C47', 'C48',
'C49', 'C50', 'C51', 'C52', 'C53', 'C54',
'C55', 'C56', 'C57', 'C58', 'C59', 'C60',
'C61', 'C62', 'C63', 'C64'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table s_et;
create table s_et(
   s_suppkey        number ,
   s_name          char(25) ,
   s_address       varchar(40) ,
   s_nationkey      number ,
   s_phone         char(15) ,
   s_acctbal       number ,
   s_comment        varchar(101)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'S1', 'S2', 'S3', 'S4', 'S5', 'S6',
'S7', 'S8', 'S9', 'S10', 'S11', 'S12',
'S13', 'S14', 'S15', 'S16', 'S17', 'S18',
'S19', 'S20', 'S21', 'S22', 'S23', 'S24',
'S25', 'S26', 'S27', 'S28', 'S29', 'S30',
'S31', 'S32', 'S33', 'S34', 'S35', 'S36',
'S37', 'S38', 'S39', 'S40', 'S41', 'S42',
'S43', 'S44', 'S45', 'S46', 'S47', 'S48',
'S49', 'S50', 'S51', 'S52', 'S53', 'S54',
'S55', 'S56', 'S57', 'S58', 'S59', 'S60',
'S61', 'S62', 'S63', 'S64'
            ))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table n_et;
create table n_et(
   n_nationkey      number ,
   n_name          char(25) ,
   n_regionkey      number ,
   n_comment        varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
            'N1'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table r_et;
create table r_et(
   r_regionkey      number ,
   r_name          char(25) ,
   r_comment        varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        nobadfile
                        nologfile
                        fields terminated by '|'
                        missing field values are null
            )
            location (
            'R1'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
alter table l_et parallel;
alter table o_et parallel;
alter table ps_et parallel;
alter table p_et parallel;
alter table c_et parallel;
alter table s_et parallel;
}
# altering tpch's default and temporary tablespace
*sql
{
alter user tpch default tablespace ts_default;
alter user tpch temporary tablespace ts_temp;
}
*sql
{
```

```
connect tpch/tpch
@?/rdbms/admin/utlxplan.sql;
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table orders;
create table orders(
   o_orderdate        ,
   o_orderkey         NOT NULL,
   o_custkey          NOT NULL,
   o_orderpriority    ,
   o_shippriority     ,
   o_clerk            ,
   o_orderstatus      ,
   o_totalprice       ,
   o_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 128m next 20m freelist groups 8 freelists 99)
compress
parallel  64
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 64
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8,
ts_lo9, ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
```

```
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
```

```
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
```

```
,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
```

```
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
```

```
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
```

```
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
```

```
,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition ord84 values less than (MAXVALUE)
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
```

```
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
)
as select
   o_orderdate         ,
   o_orderkey          ,
   o_custkey           ,
   o_orderpriority     ,
   o_shippriority      ,
   o_clerk             ,
   o_orderstatus       ,
   o_totalprice        ,
   o_comment
from o_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date

rem drop table partsupp;
create table partsupp(
   ps_partkey          NOT NULL,
   ps_suppkey          NOT NULL,
   ps_supplycost       NOT NULL,
   ps_availqty         ,
   ps_comment          ,
constraint pk_partkey_suppkey_1 primary key(ps_partkey, ps_suppkey)
)
organization index
partition by hash(ps_partkey)
partitions 64
store in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
storage (initial 1024m next 256m)
parallel  64
nologging
pctthreshold 50
as select
   ps_partkey          ,
   ps_suppkey          ,
   ps_supplycost       ,
   ps_availqty         ,
   ps_comment
from ps_et;
!date
}
*wait
*sql
{
set timing on
set echo on
!date
connect tpch/tpch;
rem drop table lineitem;
create table lineitem(
   l_shipdate          ,
   l_orderkey          NOT NULL,
   l_discount          NOT NULL,
   l_extendedprice     NOT NULL,
   l_suppkey           NOT NULL,
   l_quantity          NOT NULL,
   l_returnflag        ,
   l_partkey           NOT NULL,
   l_linestatus        ,
   l_tax               NOT NULL,
   l_commitdate        ,
   l_receiptdate       ,
   l_shipmode          ,
   l_linenumber        NOT NULL,
   l_shipinstruct      ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 512m next 50m freelist groups 8 freelists 99)
compress
parallel  64
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 64
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-
DD'))
store
```

```
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
```

```
partition item15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
```

```
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item36 values less than (to_date('1994-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
```

```
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-
DD'))
store
```

```
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
```

```
partition item58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
```

```
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
```

```
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 )
,
partition item84 values less than (MAXVALUE)
store
in (ts_lo1, ts_lo2, ts_lo3, ts_lo4, ts_lo5, ts_lo6, ts_lo7, ts_lo8, ts_lo9,
ts_lo10, ts_lo11, ts_lo12, ts_lo13, ts_lo14, ts_lo15,
ts_lo16,ts_lo17,ts_lo18,ts_lo19, ts_lo20, ts_lo21,ts_lo22, ts_lo23,
ts_lo24, ts_lo25, ts_lo26, ts_lo27, ts_lo28, ts_lo29, ts_lo30,ts_lo31,
ts_lo32, ts_lo33, ts_lo34, ts_lo35, ts_lo36, ts_lo37, ts_lo38, ts_lo39,
ts_lo40, ts_lo41, ts_lo42, ts_lo43, ts_lo44,ts_lo45,ts_lo46, ts_lo47,
ts_lo48, ts_lo49, ts_lo50, ts_lo51, ts_lo52, ts_lo53, ts_lo54, ts_lo55,
ts_lo56, ts_lo57, ts_lo58, ts_lo59,ts_lo60,ts_lo61, ts_lo62, ts_lo63,
ts_lo64 ))
as select
   l_shipdate      ,
   l_orderkey      ,
   l_discount      ,
   l_extendedprice ,
   l_suppkey       ,
   l_quantity      ,
   l_returnflag    ,
   l_partkey       ,
   l_linestatus    ,
   l_tax           ,
   l_commitdate    ,
   l_receiptdate   ,
   l_shipmode      ,
   l_linenumber    ,
   l_shipinstruct  ,
   l_comment
from l_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
rem drop table customer;
create table customer(
   c_custkey       NOT NULL,
   c_mktsegment    ,
   c_nationkey     ,
   c_name          ,
   c_address       ,
   c_phone         ,
   c_acctbal       ,
   c_comment
)
pctfree 0
pctused 99
parallel  64
nologging
storage (initial 600m next 20m)
compress
partition by hash (c_custkey)
partitions 64
store
in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
as select
   c_custkey       ,
   c_mktsegment    ,
   c_nationkey     ,
   c_name          ,
   c_address       ,
   c_phone         ,
   c_acctbal       ,
   c_comment
from c_et;
!date
}
*wait
```

```
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
rem drop table part;

create table part(
   p_partkey          NOT NULL,
   p_type             ,
   p_size             ,
   p_brand            ,
   p_name             ,
   p_container        ,
   p_mfgr             ,
   p_retailprice      ,
   p_comment
)
pctfree 0
pctused 99
parallel  64
nologging
storage (initial 700m next 20m)
compress
partition by hash (p_partkey)
partitions 64
store
in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
as select
   p_partkey          ,
   p_type             ,
   p_size             ,
   p_brand            ,
   p_name             ,
   p_container        ,
   p_mfgr             ,
   p_retailprice      ,
   p_comment
from p_et;
!date
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
rem drop table supplier;
create table supplier(
   s_suppkey          NOT NULL,
   s_nationkey        ,
   s_comment          ,
   s_name             ,
   s_address          ,
   s_phone            ,
   s_acctbal
)
pctfree 0
pctused 99
parallel  64
nologging
storage (initial 64m next 5m)
compress
partition by hash (s_suppkey)
partitions 64
store
in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
as select
   s_suppkey          ,
   s_nationkey        ,
   s_comment          ,
   s_name             ,
   s_address          ,
   s_phone            ,
   s_acctbal
from s_et;
}
*wait
*sql
{
connect tpch/tpch;
set echo on
set timing on

rem drop table nation;
create table nation(
   n_nationkey        NOT NULL,
   n_name             ,
   n_regionkey        ,
   n_comment          )
tablespace ts_default
as select * from n_et;

rem drop table region;
create table region(
   r_regionkey        ,
   r_name             ,
   r_comment          )
tablespace ts_default
as select * from r_et;
}

*wait
*bgoff
%e-scuto

*sql
{
connect tpch/tpch
set timing on
set echo on
!date
drop table l_et;
drop table o_et;
drop table ps_et;
drop table p_et;
drop table c_et;
drop table s_et;
drop table n_et;
drop table r_et;
```

```
}
*bgoff
%e-dapop
%b-ixcre
*bgon=1
############################################################
##################
# Index Creation Phase
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on
rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey) global partition by hash (l_orderkey)
partitions 64
store in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
pctfree 2
initrans 10
storage (initial 512m next 512m pctincrease 0 freelist groups 8 freelists
99)
parallel  64
compute statistics
nologging;
alter index i_l_orderkey allocate extent (size 1024m);
alter index i_l_orderkey allocate extent (size 1024m);
alter index i_l_orderkey allocate extent (size 1024m);
alter index i_l_orderkey allocate extent (size 1024m);
}
*wait
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on
rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash (o_orderkey)
partitions 64
store in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
pctfree 2
initrans 10
storage (initial 512m next 512m pctincrease 0 freelist groups 8 freelists
99 )
parallel  64
compute statistics
nologging;
alter index i_o_orderkey allocate extent (size 1024m);
alter index i_o_orderkey allocate extent (size 1024m);
}
*wait
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on

rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey) global partition by hash (c_custkey)
partitions 64
store in (ts_rest1, ts_rest2, ts_rest3, ts_rest4, ts_rest5, ts_rest6, ts_rest7,
ts_rest8, ts_rest9, ts_rest10, ts_rest11, ts_rest12, ts_rest13, ts_rest14,
ts_rest15, ts_rest16,ts_rest17,ts_rest18,ts_rest19, ts_rest20,
ts_rest21,ts_rest22, ts_rest23, ts_rest24, ts_rest25, ts_rest26, ts_rest27,
ts_rest28, ts_rest29, ts_rest30,ts_rest31, ts_rest32, ts_rest33, ts_rest34,
ts_rest35, ts_rest36, ts_rest37, ts_rest38, ts_rest39, ts_rest40, ts_rest41,
ts_rest42, ts_rest43, ts_rest44,ts_rest45,ts_rest46, ts_rest47, ts_rest48,
ts_rest49, ts_rest50, ts_rest51, ts_rest52, ts_rest53, ts_rest54, ts_rest55,
ts_rest56, ts_rest57, ts_rest58, ts_rest59,ts_rest60,ts_rest61, ts_rest62,
ts_rest63, ts_rest64 )
pctfree 2
initrans 10
storage ( initial 512m next 512m pctincrease 0 freelist groups 8 freelists
99)
parallel  64
compute statistics
nologging;
}
*wait
*bgoff
%e-ixcre
%b-anlyz
*bgon=1
############################################################
##################
# Analyze Phase
*sql
{
connect tpch/tpch;
!date
set timing on
execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent =>
1, degree => 64 , granularity => 'GLOBAL' );
connect / as sysdba
execute dbms_stats.gather_system_stats;
alter system switch logfile;
!date
}
*wait
*bgoff
%e-anlyz
```

_____

bumpx.pl

_____

```
#!/usr/local/bin/perl
#
# $Header: bumpxlite.pl 23-oct-2002.13:15:45 mpoess Exp $
#
# bumpxlite.pl
#
# Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.
#
#    NAME
#      bumpxlite.pl - <one-line expansion of the name>
#
#    DESCRIPTION
```

```
#     <short description of component this file declares/defines>
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      10/23/02 - mpoess_update_from_visa
#   mpoess      09/24/01 - take out readfile subroutine
#   mpoess      08/10/01 - Creation
#

$os = $ENV{'OS'};
if (($os cmp 'Windows_NT') != 0) { # os is UNIX
   $os = "unix";$nt = 0;$unix = 1;
} else {
   $os = "nt"; $nt = 1; $unix = 0;
}
$| = 1;
$verbose = 0;
if (($os cmp "unix")==0) {
   $defphases = "dbcre,sctso,scuto,dbgen,dapop,anlyz,ixcre";
} else {
   $defphases =
"sdgen,shutd,start,dbgen,plcre,dbcre,sctso,scuto,dapop,scuvo,anlyz,ixcre
,chob";
}
$allbmtypes = "tpcd,wisc";
$bmtype = "tpcd" if !defined $bmtype;
$pdfile =  "$ENV{'BUMPX_DIR'}/param.txt";  # This file contains the
description of all possible parameters.
while ($arg = shift(@ARGV)) {
   if ($arg !~ /-(i|o|t|p|d|a|s|h)/){
      $error =  "** Error: Bad argument to $0: $arg\n";
      &usage;
   }
   if ($arg =~ /-h/) { &usage;ext(0);}
   $runsilent = 1 if ($arg =~ /-s/);
   $outfile = shift(@ARGV) if ($arg =~ /-o/);
   $bmtype = shift(@ARGV) if ($arg =~ /-t/);
   $phaselist = shift(@ARGV) if ($arg =~ /-p/);
   if ($arg =~ /-d/) {
            $defpar = shift(@ARGV);
            @keys = keys %params;
            while ($#keys >=0) {
               $key = pop(@keys);
               if (($defpar cmp "") ==0) {
                        print $key, "=", $params{$key}, "\n";
               } else {
                        print $key, "=", $params{$key}, "\n" if ($key =~
/$defpar/);
               }
            }
            exit(0);
   }
}
$outfile = "$ENV{'BUMPX_DIR'}/bumpx.dat" if !defined $outfile;
if ($nt) {
   $listdir = $filedir."list/";
   if (!-e $listfile) {
            system ("mkdir $listdir");
   }
}
if (($os cmp "nt") == 0) {          ## NT Port (Use tmpfile to buffer
   $tmpfile = "tmp.txt";       ## commands and nruntpb to synchronize
them)
   $tmpfile = $filedir.$tmpfile;
   $nruntpb =  "nruntpb.exe";
} ## NT End
if (!-e $outfile) {
   $error = "** Error: -o file, $outfile, does not exist\n";
   &usage;
}
$phaselist = $defphases if !defined $phaselist;
@phases = split(/,/, $phaselist);
## NT Port (Use tmpfile to buffer commands for nruntpb)
open (TMPFILE, ">$tmpfile") if ( (($os cmp "nt") == 0));
## NT End
&doexecute;
## NT Port
close(TMPFILE) if ( (($os cmp "nt") == 0));
## NT End
exit(0);

sub doexecute { # First, do preprocessing stuff
   print "Execution pass begun." if $verbose;
   open (INFILE, $outfile);
  WLOOP1:
   while ($line = <INFILE>)
   {
      study $line;
      next WLOOP1 if $line =~ /^\s*\#/;
      next WLOOP1 if $line =~ /^\s*\n/;
      if ($line =~ /^%b-preproc/)
      {
         $insection = 1;
         next WLOOP1;
      }
      next WLOOP1 if ($insection != 1);
      if ($line =~ /^%e-preproc/)
      {
         $insection = 0;
         $commands{$shortcmd} = $longcmd if defined $shortcmd;
         last WLOOP1;
      }
      if ($line =~ /^\*/)
      {
         $commands{$shortcmd} = $longcmd if defined $shortcmd;
         $line =~ /^(\*.*\S+)\s*\n$/;
         $shortcmd = $1;
         $longcmd = "";
         next WLOOP1;
      }
      if ($line =~ /^\\/)
      {
         #$line =~ /\\(.*\n)/;
         $line =~ /^\\(.*\n)/;
         $longcmd = $longcmd . $1;
         next WLOOP1;
      }
      print "Illegal entry in preproc stage:\n   $line";
   }
   close (INFILE);

# Then, do all of the requested phases
   $execctr = 0;
   foreach $phase (@phases)
   {
            $phase_cmd_num = 0;
            print "\n  Executing phase \'$phase\'" if $verbose;
            $bg = 0;
     open (INFILE, $outfile);
    WLOOP2:
     while ($line = <INFILE>)
     {
        study $line;
        next WLOOP2 if $line =~ /^\s*\#/;
        next WLOOP2 if $line =~ /^\s*\n/;
        if ($line =~ /^\*ignon/)
        {
           $ignon = 1;
```

```
            next WLOOP2;
        }
        if ($line =~ /^\*ignoff/)
        {
            $ignon = 0;
            next WLOOP2;
        }
        next WLOOP2 if ($ignon == 1);
        if ($line =~ /^%b-$phase/)
        {
            $insection = 1;
            $execcmd = "";
            next WLOOP2;
        }
        next WLOOP2 if ($insection != 1);
        if ($line =~ /^%e-$phase/)
        {
            $insection = 0;
            &execute ($execcmd);
            last WLOOP2;
        }
        if ($line =~ /^\*(.*)/)
        {
            &execute ($execcmd);
            if (($1 =~ /bgo/) || ($1 =~ /wait/) || ($1 =~ /ignore/))
            {
                $execcmd = $line;
                next WLOOP2;
            }
            $line =~ /^(\*.*\S+)\s*\n$/;
            $execcmd = $commands{$1};
            next WLOOP2;
        }
        if ($line =~ /^\{(.*)\}/)
        {
            $insert = "";
            $insert = $1;
            $execcmd =~ s/\{\}/$insert/;
            next WLOOP2;
        }
        if ($line =~ /^\{(.*)$/)
        {
            $insubsection = 1;
            $insert = "";
            $insert = $1;
            next WLOOP2;
        }
        if ($line =~ /^(.*)\}/)
        {
            $insubsection = 0;
            $insert = $insert . $1;
                        if (($os cmp "nt") == 0){ ## NT Port (Ignore '\n')
                            $insert =~ /(.*)\n$/s;
                            $insert = $1;
                        } ## NT End
            $execcmd =~ s/\{\}/$insert/;
            next WLOOP2;
        }
        $insert = $insert . $line if ($insubsection == 1);
    }
    close (INFILE);
  }
  print "\nExecution pass complete.\n" if $verbose;
}

sub execute
{
  $cmd = shift(@_);
  if ($cmd)
  {
    return if ($cmd =~ /^\*ignore/);
    if ($cmd =~ /^\*bgon=(.*)/)
    {
        $bgmax = $1;
        $bg = 1;
        $bgrun = 0;
        return;
    }
    if ($cmd =~ /^\*bgoff/)
    {
        $bg = 0;
        return;
    }

            if ($cmd =~ /^\*time=(.*)/)  ##NT only
    {
                print $1 . "\n";
                print localtime(time) . "\n";
        return;
    }
    if ($cmd =~ /^copy (.*)/) ## NT only
    {
                system($cmd);
        ## Quit if failed
        if ($?) {
                            print "system copy command
failed:\n$cmd\nreason: $? ($!)\n";
                    exit(-1);
                }
        return;
    }
            if ($cmd =~ /^del (.*)/) ## NT only
    {
                system($cmd);
        ## Quit if failed
        if ($?) {
                            print "system del command
failed:\n$cmd\nreason: $? ($!)\n";
                    exit(-1);
                }
        return;
    }

                if ($cmd =~ /^\*wait/) ## This deals with main differences
between NT and UNIX
    {
                if (($os cmp "unix") == 0)
                {
                            while ($fpid = shift(@wpids))
                            {
                                waitpid($fpid, 0);
                            }
                }
                else
                {    ## NT Port (Start background tasks if any. nruntpb
will wait until all tasks are done)
                            if ($bgrun >= 1)
                            {
                                close(TMPFILE);
                                system("cat $tmpfile >> $listdir$phase.lst");
                                system("vi $tmpfile") if $debug;
                                system("$nruntpb -p < $tmpfile") if !$debug;
                                if ($?)
                                {
                                                print "system command
failed:\n$nruntpb < $tmpfile\n";
                                                print "reason: $? ($!)\n";
                                                print "Please check the contents in the
input file.\n";
                                                exit(-1);
```

```
                        }
                        open(TMPFILE, ">$tmpfile");
                    }
                }
            $bgrun = 0;
            return;
        }
        if ($cmd =~ /(s|g)etenv/)
        {
                @lines = split(/\n/, $cmd);
            $cmd = "";
            foreach $line (@lines)
            {
                while (1)
                {
                    last if ($line !~ /getenv/);
                    $line =~ /(.*)\*getenv\(([^(\)|\*)]*)\)(.*)/;
                    $line = $1 . $ENV{$2} . $3;
                }
                if ($line =~ /jojo/) #we do not want to use this for now
                {
                    $line =~ /setenv\s+(\S+)\s+(\S+)/;
                    $ENV{$1} = $2;
                }
                else
                {
                    $cmd = $cmd . $line. "\n";
                }
            }
        }
        return if ($cmd !~ /\S+/); # return if nothing left to execute
            $execctr++;
        $ENV{'BUMPX_CTR'} = $$.'-'.$execctr;
            if (($os cmp "unix") == 0)
            {
                if ($bg == 1)
                {
                        print "." if $verbose;
                        $fpid = fork;
                        if ($fpid == 0)
                        {
                            exec ($cmd);
                            print "exec\'d command failed:\n$cmd\nreason:
$!\n";
                            exit(-1);
                        }
                        unshift (@wpids, $fpid);
                        $bgrun = $bgrun + 1;
                        &execute ("*wait") if (($bgrun >= $bgmax) &&
($bgmax >= 0));
                }
                else
                {
                        system ($cmd);
                        print "system\'d command failed:\n$cmd\nreason:
$? ($!)\n" if $?;
                }
            }
            else ## NT support
            {
        ## NT Port (Submit background tasks if there are bgmax of them,
otherwise write to tmpfile)
                if ($bg == 1)
                {
                        print "." if $verbose;
                        if ($bgrun < $bgmax)
                        {
                            $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                            ++$phase_cmd_num;
                            print TMPFILE $cmd;
                            $bgrun = $bgrun + 1;
                        }
                        else
                        {
                            close(TMPFILE);
                            system("cat $tmpfile >> $listdir$phase.lst");
                            system("$nruntpb -p < $tmpfile");
                            if ($?) {
                                print "system command
failed:\n$nruntpb < $tmpfile\nreason: $? ($!)\n";
                                print "Please check the contents in the
input file.\n";
                                exit(-1);
                            }
                            open(TMPFILE, ">$tmpfile");
                            $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                            ++$phase_cmd_num;
                            print TMPFILE $cmd;
                            $bgrun = 1;
                        }
                    }
                    else
                    {
                            $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                            ++$phase_cmd_num;
                            print TMPFILE $cmd;
                            close(TMPFILE);
                            system("cat $tmpfile >> $listdir$phase.lst");
                            system ("sh $tmpfile");
                            if ($?) {
                                print "system\'d command failed:\nsh
$tmpfile\nreason: $? ($!)\n";
                                print "Please check the contents in the shell
script.\n";
                                exit(-1);
                            }
                            open(TMPFILE, ">$tmpfile");
                    }
                }  ## NT support End
    }
}

sub usage
{
    print "Usage:\n";
    print "This is a lite version of bumpx.pl.  It can only be used to
execute a .dat file\n";
    print " $0 [-o outfile] [-p phaselist] [-t type]\n";
    print "   -o : intermediary file to be created and/or used\n";
    print "        defaults to bumpx.dat in \$BUMPX_DIR or \$CWD\n";
    print "   -p : list of phases to create/execute\n";
    print "        phaselist is a comma separated list of phases in order\n";
    print "        possible phases are:\n";
    print "          sdgen = seed file generation\n";
    print "          dbgen = data flat file generation\n";
    print "          plcre = NT raw partition and links creation\n";
    print "          dbcre = database creation\n";
    print "          shutd = shutdown database (on all instances)\n";
    print "          start = startup database (on all instances)\n";
    print "          sccre = schema creation\n";
    print "          sctso = schema creation (tablespaces only)\n";
    print "          scuto = schema creation (user and tables only)\n";
    print "          scuvo = schema creation (views only)\n";
    print "          dapop = data population\n";
    print "          ixcre = index creation (including constraints)\n";
    print "          anlyz = analyze objects\n";
    print "          chob  = change parameters of objects\n";
```

```
print "          expln = create explain plans\n";
print "          query = run and time queries\n";
print "        defaults to $defphases\n";
print "   -t : type of benchmark\n";
print "        enables benchmark-specific defaults\n";
print "        current possibilties are: $allbmtypes\n";
print "        defaults to tpcd\n";
print "   -s : run silent (no parameter checking is done)\n";
print "\n";
    print "Examples:\n";
    print " $0 -p dapop\n";
    print "    Executes data population phase of intermediary file
bumpx.dat.\n";
    print "\n";
    print "$error\n";
    exit(-1);
}
```

Appendix C: ACID Scripts

_____
a_query2.sql
_____

```
Rem
Rem $Header: aquery2.sql 07-aug-99.23:54:47 mpoess Exp $
Rem
Rem aquery2.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      aquery2.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Performs query on PARTSUPP for TPC-D benchmark
Rem       Isolation Test 5.
Rem       Asks user to input values for ps_partkey and ps_suppkey
Rem       The range for ps_partkey is 1 to 20000
Rem       The range for ps_suppkey is 1 to 1000
Rem       A valid combination is 46 and 47
Rem  Usage:   sqlplus tpcd/tpcd @a_query2 <ps_partkey>
<ps_suppkey>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/07/99 - Creation
Rem    mpoess      08/07/99 - Created
Rem
rem  DESCRIPTION
rem       Performs query on PARTSUPP for TPC-D benchmark
rem       Isolation Test 5.
rem       Asks user to input values for ps_partkey and ps_suppkey
rem       The range for ps_partkey is 1 to 20000
rem       The range for ps_suppkey is 1 to 1000
rem       A valid combination is 46 and 47

set serverout on;

select
'BEFORE PARTSUPP QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as
CURRENT_TIME
from dual;

select *
from partsupp
where ps_partkey = &&1
and ps_suppkey = &&2;

select
'AFTER PARTSUPP QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as
CURRENT_TIME
from dual;

exit;
```

_____
a_query.sql
_____

```
Rem
Rem $Header: a_query.sql 06-aug-99.10:51:10 mpoess Exp $
Rem
Rem a_query.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      a_query.sql - <one-line expansion of the name>
Rem
rem  DESCRIPTION
Rem       Performs ACID Query for TPC-D benchmark.
Rem       Asks user to input values for o_key
Rem       The range of okey is 1 to 600000
Rem
================================================
===============
Rem
Rem Usage:   sqlplus tpcd/tpcd @a_query <o_key>
Rem
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/06/99 - Creation
Rem    mpoess      08/06/99 - Created
Rem

set serverout on;

select
'BEFORE ACID QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as
CURRENT_TIME
from dual;

select SUM(trunc(trunc(l_extendedprice * (1-l_discount),2) *
(1+l_tax),2)) AS RESULT
from lineitem
where l_orderkey = &&1;

select
'AFTER ACID QUERY' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as
CURRENT_TIME
from dual;

exit;
```

_____
atom.sh
_____

```
#!/bin/ksh
#
# $Header: atom.sh 08-aug-99.13:48:02 mpoess Exp $
#
# atom.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      atom.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Performs atomicity tests.
#      Usage: atom.sh [-n iter] [-p prog] [-u usr/pswd] -h
#
```

```
#     Options: See usage below
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      08/08/99 - Creation
#   mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit set in env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura

usage() {

  echo ""
  echo "Usage: $0 [-n iter] [-p prog] [-u usr/pswd] -h"
  echo ""
  echo "-n iter     : number of iterations, default is 100"
  echo "-p prog     : program to run, default is atranspl.ott"
  echo "-u usr/pswd : user/password combo for database access, default
is tpcd/tpcd"
  echo "-h          : print this usage summary"
  exit 1;
}

ITER=3
SF=1
PROG=$KIT_DIR/utils/atranspl
OUT=${OUT_DIR}/atom
USER=${DATABASE_USER}

set -- `getopt "n:p:u:h" "$@"` || usage

while :
do
  case "$1" in
  -n) shift; ITER=$1;;
  -p) shift; PROG=$1;;
  -u) shift; USER=$1;;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift
done

echo "Starting Atomicity Test at `date`..."
echo ""
echo "Performing $ITER ACID transactions with COMMIT"
echo ""

$KIT_DIR/utils/randkey $ITER $SF u$USER | $PROG 1 1 1 0 u$USER
> ${OUT}c 2>&1

echo "ACID transactions with COMMIT ended. Output in ${OUT}c"
echo ""
echo "Performing $ITER ACID transactions with ROLLBACK"
echo ""

$KIT_DIR/utils/randkey $ITER $SF u$USER | $PROG 1 1 0 0 u$USER
> ${OUT}r 2>&1

echo "ACID transactions with ROLLBACK ended. Output in ${OUT}r"
echo ""
echo "Ending Atomicity Test at `date`..."
```

atranspl.c

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*

  NAME
    atranspl.c - <one-line expansion of the name>

  DESCRIPTION
    TPC-HR benchmark ACID transaction driver, OCI version 8

  NOTES
    <other useful comments, qualifications, etc.>

  MODIFIED   (MM/DD/YY)
  mpoess      10/23/02 - mpoess_update_from_visa
  mpoess      10/17/01 - add parameter in ACIDinit
  mpoess      02/22/01 - enlarge timing array
  mpoess      01/04/01 - Creation

*/

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#include "atranspl.h"

/* Declare error handling functions */

double gettime();
void sql_error();
void usage();
void ACIDinit();
void ACIDexit();
int atoi();
void srand48();
long lrand48();

/* declarations for ORDERS */

int o_key = 0;
double o_tprice = 0.0;
double o_newtprice = 0.0;

/* declarations for LINEITEM */

int l_key = 0;
int l_pkey = 0;
int l_skey = 0;

int l_quan = 0;
int l_newquan = 0;
double l_eprice = 0.0;
double l_neweprice = 0.0;
double l_disc = 0.0;
double l_tax = 0.0;

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;
```

```
int proc_no = 1;          /* process number, global             */
int num_streams = 1;      /* number of transaction streams      */
int trig = 0;             /* Trigger Time                       */
int slp = 0;              /* Sleep Time                         */

int logfile;              /* fdes for logfile for durability (optional)   */
int outfile = 1;          /* output file (optional)             */
#ifdef LINUX
  FILE *infile;     /* input file (optional)              */
#else
  FILE *infile = stdin;     /* input file (optional)              */
                    /* in the format of <o_key> <delta>    */
#endif
char lname[UNAME_LEN];    /* username/passwd combo              */
char *passwd;             /* pointer to password                */

char buf[WRITE_BUF_LEN];  /* buffer to write                    */

unsigned flag = (unsigned) 0;       /* flag to store all sorts of options  */

#define INFILE  0x01u
#define OUTFILE 0x02u
#define LOGFILE 0x04u
#define COMMIT  0x08u
#define DELTA   0x10u

double tr_end = 0.0;      /* transaction end time               */
double tr_start = 0.0;    /* transaction start time             */

int num_iter = 0;         /* number of iterations               */

time_t curr_time;         /* Current Time                       */

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curi = NULL;
OCIStmt *curr = NULL;
OCIStmt *cure1 = NULL;
OCIStmt *cure2 = NULL;

/* OCI bind handles */

#ifdef NOLKEY
OCIBind *l_keyi_bp = NULL;
OCIBind *o_keyi_bp = NULL;
#endif /* NOLKEY */

OCIBind *l_key_bp = NULL;
OCIBind *o_key_bp = NULL;
OCIBind *delta_bp = NULL;
OCIBind *l_pkey_bp = NULL;
OCIBind *l_skey_bp = NULL;
OCIBind *l_quan_bp = NULL;
OCIBind *l_newquan_bp = NULL;
OCIBind *l_tax_bp = NULL;
OCIBind *l_disc_bp = NULL;
OCIBind *l_eprice_bp = NULL;
OCIBind *l_neweprice_bp = NULL;
OCIBind *o_tprice_bp = NULL;
OCIBind *o_newtprice_bp = NULL;
OCIBind *rprice_bp = NULL;
OCIBind *cost_bp = NULL;

OCIBind *l_neweprice1_bp = NULL;
OCIBind *l_newquan1_bp = NULL;
OCIBind *o_key1_bp = NULL;
OCIBind *l_key1_bp = NULL;

OCIBind *o_newtprice2_bp = NULL;
OCIBind *o_key2_bp = NULL;

sword status = OCI_SUCCESS;  /* OCI return value */

char sqlstmt[1024];

/* usage: prints the usage of the program */

void usage()
{

  fprintf (stderr,"\nUsage: atrans.o[st]t <proc_no> <num_streams>
<commit> <delta>\n[i<pathname for input>] [o<pathname for output>]
[d<pathname for durability file>] [u<uid/passwd>] \n\n");

  fprintf(stderr,"    proc_no      :the process number within this ACID\n");
  fprintf(stderr,"    num_streams  :the total number of ACID transaction
streams\n");
  fprintf(stderr,"    commit       :1 to commit transaction, abort
otherwise\n\n");
  fprintf(stderr,"    delta        :1 to generate new random delta, otherwise
obtain delta from input\n\n");
  fprintf(stderr,"    OPTIONAL PARAMETERS:\n");
  fprintf(stderr,"    i<pathname for input>      :full path name for input file
- default is stdin\n");
  fprintf(stderr,"    o<pathname for output>     :full path name for output
file - default is stdout\n");
  fprintf(stderr,"    d<pathname for durability> :full path name for
durability success file - must specify for durability test\n");
  fprintf(stderr,"    u<uid/passwd>             :Username/Password string -
default is tcpd/tpcd\n");
  fprintf(stderr,"    t<trigger>                :Trigger Time - sleep <trigger>
seconds before start\n\n");
  fprintf(stderr,"    s<sleep>                  :Sleep Time - sleep <sleep>
seconds before commit or rollback\n\n");
  exit(-1);

}


void ACIDexit() {

  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);

}



/* type: 0 if environment handle is passed, 1 if error handle is passwd */

void sql_error(errhp,status,type)
    OCIError *errhp;
    sword status;
    sword type;
{
  char msg[2048];
  ub4 errcode;
  ub4 msglen;
  int i,j;

  switch(status) {
```

```
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement returned with info.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*) &errcode, (text*) msg,
                  2048, OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*) &errcode, (text*) msg,
                  2048, OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_ERROR:
    fprintf(stderr, "Error: OCI call error.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
                  2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
                  2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
                  2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL, (sb4 *) &errcode, (text*) msg,
                  2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }
  /* Rollback just in case */

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

  fprintf(stderr, "Exiting Oracle...\n");
  fflush(stderr);

  ACIDexit();

  exit(1);
}

#ifdef LINUX
int main(argc,argv)
#else
void main(argc,argv)
#endif
     int argc;
     char *argv[];
{

  int i;
  char line[64];
  ub4 errcode;
  char msg[2048];
  int need_commit = 0;

  /* Initialize some variables */
#ifdef LINUX
  infile=fopen("/dev/stdin","r");
#endif
  strcpy((char *) lname, "tpcd/tpcd");

  if ((argc > 10) || (argc < 5)) {
    usage();
  }

  /* argv[1] -- Process Number */

  proc_no = atoi(argv[1]);

  /* argv[2] -- Number of Streams */

  num_streams = atoi(argv[2]);

  /* argv[3] -- Commit? */

  if (atoi(argv[3]) == 1)
    BIS(flag, COMMIT);

  /* argv[4] -- Delta? */

  if (atoi(argv[4]) == 1)
    BIS(flag, DELTA);

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'u':
      strncpy((char *) lname, ++(argv[0]), UNAME_LEN);
      if (strchr((char *) lname, '/') == NULL) {
        fprintf(stderr, "Login name must be in the format of
userid/passwd\n");
        usage();
        exit(-1);
      }
      break;
    case 'i':
      if ((infile = fopen(++(argv[0]), "r")) == NULL) {
        fprintf(stderr,"Cannot open input file %s\n", argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, INFILE);
      break;
    case 'o':
      if ((outfile = open(++(argv[0]), (O_RDWR | O_SYNC | O_CREAT),
S_IRWXU)) == -1) {
        fprintf(stderr,"Cannot open output file %s\n", argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, OUTFILE);
      break;
    case 'd':
      if ((logfile = open(++(argv[0]), (O_RDWR | O_SYNC | O_CREAT),
S_IRWXU)) == -1) {
        fprintf(stderr,"Cannot open durability success file %s\n", argv[0]);
        fprintf(stderr,"%s\n",strerror(errno));
        exit(-1);
      }
      BIS(flag, LOGFILE);
      break;
    case 'b':
      num_iter = atoi(++(argv[0]));
      break;
    case 't':
      trig = atoi(++(argv[0]));
      break;
    case 's':
      slp = atoi(++(argv[0]));
      break;
    default:
      fprintf(stderr, "Unknown argument %s\n", argv[0]);
```

```
    usage();
    break;
   }
  }

  FPRTF(outfile,"-----------------------------------------------\n");

  /* Initialize the cursors etc. */

  (void) ACIDinit();

  /* sleep for some time (triggering) */

  sleep(trig);

  /* start doing the ACID transactions */

  tr_start = gettime();

  /* The number of iteration we will run depends on the number of */
  /* input lines                                                  */

  while (fgets(line, 64, infile) != NULL) {

#ifdef NOLKEY
    sscanf(line, "%d %d\n", &o_key, &delta);

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

    /* l_key is the highest l_linenumber available.  We need to pick */
    /* at random a number between 1..l_key.                          */

    l_key = (int) ((lrand48() % l_key) + 1);
#else
    sscanf(line, "%d %d %d\n", &o_key, &l_key, &delta);
#endif /* NOLKEY */

    /* Generate delta if necessary */

    if (BIT(flag, DELTA))
      delta = (int) (floor((drand48() * 100)) + 1);

    /* Now, we are ready to run the ACID transaction. */

    curr_time = time(NULL);

    FPRTF2(outfile, "Starting ACID transaction %d at %s...\n",
(++num_iter),
           ctime(&curr_time));

    FPRTF1(outfile, "o_key: %d\n", (int) o_key);
    FPRTF1(outfile, "l_key: %d\n", (int) l_key);
    FPRTF1(outfile, "delta: %d\n", (int) delta);

    OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

    if (!BIT(flag, LOGFILE)) {
    FPRTF1(outfile, "BEFORE COMMIT/ROLLBACK
TRANSACTION at %s\n", ctime(&curr_time));
    FPRTF1(outfile, "l_extendedprice: %.2f\n", l_eprice);
    FPRTF1(outfile, "l_quantity:      %d\n", (int) l_quan);
    FPRTF1(outfile, "o_totalprice:    %.2f\n\n", o_tprice);
    }

    FPRTF1(outfile, "Sleep %d seconds before
COMMIT/ROLLBACK...\n\n", slp);
    sleep(slp);

    /* Shall we commit? */

    if (BIT(flag, COMMIT)) {
      need_commit = 1;
      while (need_commit) {
        if((status=OCITransCommit(tpcsvc,errhp,OCI_DEFAULT)) !=
OCI_SUCCESS) {
            OCIrol(tpcsvc,errhp);
            OCIsexec(tpcsvc,curr,errhp,1);
        } else {
           need_commit = 0;
           curr_time = time(NULL);
           FPRTF2(outfile, "ACID Transaction iteration %d COMMITED
at %s\n",
             num_iter, ctime(&curr_time));
        }
      }
    } else {
      OCIrol(tpcsvc,errhp);
      curr_time = time(NULL);
      FPRTF2(outfile, "ACID Transaction iteration %d ROLLBACK at
%s\n",
           num_iter, ctime(&curr_time));
    }

    /* Report all results to outfile and if necessary, to success file. */

    /* Report initial and new values for o_totalprice, l_extendedprice, */
    /* l_quantity.                                                      */

    /*
    curr_time = time(NULL);
    FPRTF1(outfile, "Transaction Completed at %s\n",
ctime(&curr_time));
    */

    /* Get the values in LINEITEM and ORDERS after the transaction */

    if (BIT(flag, LOGFILE)) {
      FPRTF1(logfile, "p_key:      %d\n", (int) l_pkey);
      FPRTF1(logfile, "s_key:      %d\n", (int) l_skey);
      FPRTF1(logfile, "o_key:      %d\n", (int) o_key);
      FPRTF1(logfile, "l_key:      %d\n", (int) l_key);
      FPRTF1(logfile, "delta:      %d\n", (int) delta);
      FPRTF1(logfile, "Transaction Completed at %s\n",
ctime(&curr_time));
      FPRTF(logfile, "-----------------------------------------------\n");
    } else {

      OCIsexec(tpcsvc,cure1,errhp,1);
      OCIsexec(tpcsvc,cure2,errhp,1);

      FPRTF(outfile, "AFTER TRANSACTION:\n");
      FPRTF1(outfile, "l_extendedprice: %.2lf\n", l_neweprice);
      FPRTF1(outfile, "l_quantity:      %d\n", (int) l_newquan);
      FPRTF1(outfile, "o_totalprice:    %.2lf\n\n", o_newtprice);
      FPRTF1(outfile, "l_tax:           %.2lf\n", l_tax);
      FPRTF1(outfile, "l_discount:      %.2lf\n", l_disc);
      FPRTF1(outfile, "rprice:          %.2lf\n", rprice);
      FPRTF1(outfile, "cost:            %.2lf\n", cost);
      FPRTF(outfile, "-----------------------------------------------\n");
    }
  }

  tr_end = gettime();

  if (!BIT(flag,LOGFILE)) {
    FPRTF1(outfile, "Start Time: %.2f\n", tr_start);
```

```
    FPRTF1(outfile, "End Time: %.2f\n", tr_end);
    FPRTF1(outfile, "Elapsed Time: %.2f\n", (tr_end - tr_start));
    FPRTF1(outfile, "Transaction Count: %d\n", num_iter);
    FPRTF1(outfile, "Transaction Rate: %.2f\n", num_iter/(tr_end -
tr_start));
  } else {
    FPRTF1(logfile, "Start Time: %.2f\n", tr_start);
    FPRTF1(logfile, "End Time: %.2f\n", tr_end);
    FPRTF1(logfile, "Elapsed Time: %.2f\n", (tr_end - tr_start));
    FPRTF1(logfile, "Transaction Count: %d\n", num_iter);
  }

  /*  Disconnect from ORACLE. */

  if (BIT(flag, INFILE))
    fclose(infile);
  if (BIT(flag, OUTFILE))
    close(outfile);
  if (BIT(flag, LOGFILE))
    close(logfile);

  ACIDexit();

  exit(0);
}


void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

  /* Connect to ORACLE.  Program will call sql_error()
     if an error occurs in connecting to the default database. */

  (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);
  if((status=OCIEnvInit((OCIEnv **)&tpcenv,OCI_DEFAULT,0,(dvoid
**)0)) !=
     OCI_SUCCESS)
    sql_error(tpcenv, status, 0);

  OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
  OCIhalloc(tpcenv,&curi,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&curr,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cure1,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cure2,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
  OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

  /* Disables auto commit */
/*
  if (ocof(&tpclda)) {
    sql_error(&tpclda, &tpclda);
    ologof(&tpclda);
    exit(-1);
  }
*/

  /* get username and password */

  passwd = strchr(lname, '/');
  *passwd = '\0';
  passwd++;

  if ((status = OCIServerAttach(tpcsrv,errhp,(text
*)0,0,OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);

OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER
,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATT
R_USERNAME,
        errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_A
TTR_PASSWORD,
        errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                        OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESSIO
N,errhp);

  /* Enable session parallel dml */

  sprintf((char *) sqlstmt, PDMLTXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);

  /* Enable session parallel ddl */

  /*sprintf((char *) sqlstmt, PDDLTXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);*/


  /* Make session serializable */

  sprintf ((char *) sqlstmt, ISOTXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);

  /* Set optimizer_index_cost_adj = 25 */

  sprintf ((char *) sqlstmt, OICATXT);
  OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);
  OCIsexec(tpcsvc,curi,errhp,1);

  curr_time = time(NULL);
  printf("\nConnected to ORACLE as user: %s at %s\n\n", lname,
ctime(&curr_time));


#ifdef NOLKEY
  /* Open and Parse cursor for query to choose determine l_key. */
  /* Binds l_key to :l_key.                                     */

  sprintf((char *) sqlstmt,SQLTXT1);
  OCIStmtPrepare(curi,errhp,sqlstmt,strlen((char
*)sqlstmt),OCI_NTV_SYNTAX,OCI_DEFAULT);



OCIbbname(curi,&l_keyi_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQ
LT_INT);
```

```
OCIbbname(curi,&o_keyi_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),
SQLT_INT);

#endif /* NOLKEY */

  /* Open and Parse cursor for the ACID transaction.          */

  sprintf((char *) sqlstmt,SQLTXT2);
  OCIStmtPrepare(curr,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);

  /* bind variables */


OCIbbname(curr,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQL
T_INT);

OCIbbname(curr,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQ
LT_INT);

OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQLT_I
NT);

OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",ADR(l_pkey),SIZ(l_pkey),
SQLT_INT);

OCIbbname(curr,l_skey_bp,errhp,":l_skey",ADR(l_skey),SIZ(l_skey),S
QLT_INT);

OCIbbname(curr,l_quan_bp,errhp,":l_quan",ADR(l_quan),SIZ(l_quan),
SQLT_INT);
  OCIbbname(curr,l_newquan_bp,errhp,":l_newquan",ADR(l_newquan),
        SIZ(l_newquan),SQLT_INT);

OCIbbname(curr,l_tax_bp,errhp,":l_tax",ADR(l_tax),SIZ(l_tax),SQLT_
FLT);

OCIbbname(curr,l_disc_bp,errhp,":l_disc",ADR(l_disc),SIZ(l_disc),SQ
LT_FLT);

OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_epr
ice),
        SQLT_FLT);

OCIbbname(curr,l_neweprice_bp,errhp,":l_neweprice",ADR(l_newepric
e),
        SIZ(l_neweprice),SQLT_FLT);


OCIbbname(curr,o_tprice_bp,errhp,":o_tprice",ADR(o_tprice),SIZ(o_tp
rice),
        SQLT_FLT);

OCIbbname(curr,o_newtprice_bp,errhp,":o_newtprice",ADR(o_newtpri
ce),
        SIZ(o_newtprice), SQLT_FLT);
  OCIbbname(curr,rprice_bp,errhp,":rprice",ADR(rprice),SIZ(rprice),
SQLT_FLT);
  OCIbbname(curr,cost_bp,errhp,":cost",ADR(cost),SIZ(cost),
SQLT_FLT);

  /* Open & Parse cursor for end values query */

  sprintf((char *) sqlstmt,SQLTXT3);
  OCIStmtPrepare(cure1,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);

  sprintf((char *) sqlstmt,SQLTXT4);
  OCIStmtPrepare(cure2,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);

  /* bind variables */


OCIbbname(cure1,l_neweprice1_bp,errhp,":l_neweprice",ADR(l_newep
rice),
        SIZ(l_neweprice),SQLT_FLT);

OCIbbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_newqua
n),
        SIZ(l_newquan),SQLT_INT);

OCIbbname(cure1,o_key1_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),
SQLT_INT);

OCIbbname(cure1,l_key1_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),S
QLT_INT);


OCIbbname(cure2,o_newtprice2_bp,errhp,":o_newtprice",ADR(o_newt
price),
        SIZ(o_newtprice),SQLT_FLT);

OCIbbname(cure2,o_key2_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),
SQLT_INT);

}
```

atranspl.h

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*
   NAME
     atranspl.h - <one-line expansion of the name>

   DESCRIPTION

   MODIFIED   (MM/DD/YY)
   mpoess      10/23/02 - mpoess_update_from_visa
   mpoess      10/17/01 - add TXT parameter
   mpoess      04/09/01 - add hint to find max linenumber
   mpoess      01/04/01 - Creation

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/param.h>
#include <sys/types.h>
#include <time.h>
#include <errno.h>
#include <math.h>

#include <oratypes.h>
#ifndef OCIDFN
#include <ocidfn.h>
#endif /* OCIDFN */

#ifndef OCI_ORACLE
#include <oci.h>
#endif /* OCI_ORACLE */
```

```
/*
#ifdef __STDC__
#include <ociapr.h>
#else
#include <ocikpr.h>
#endif */ /* __STDC__ */

extern int errno;

#ifndef NULL
#define NULL 0
#endif

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024

#define NA              -1      /* ANSI SQL NULL */
#define VER7            2
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not
serializable */
#define WRITE_BUF_LEN 1024

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define BIS(flg,mask) (unsigned) (flg |= (unsigned) mask)
#define BIT(flg,mask) (unsigned) ((unsigned) flg & (unsigned) mask)

#define FPRTF(fd,s) \
{sprintf(buf,s); write(fd, buf, strlen(s));}
#define FPRTF1(fd,s,p) \
{sprintf(buf,s,p); write(fd, buf, strlen(buf));}
#define FPRTF2(fd,s,p1,p2) \
{sprintf(buf,s,p1,p2); write(fd, buf, strlen(buf));}

#define OCIhalloc(envh,hndl,htyp) \
   if((status=OCIHandleAlloc((dvoid *)envh,(dvoid
**)hndl,htyp,0,(dvoid **)0))!=OCI_SUCCESS) \
      sql_error(envh,status,0); \
   else \
      DISCARD 0

#define OCIhfree(hndl,htyp) \
   if((status=OCIHandleFree((dvoid *)hndl,htyp)) == OCI_SUCCESS) \
      fprintf(stderr, "Error freeing handle of type %d\n", htyp)

#define OCIaget(hndl,htyp,attp,size,atyp,errh) \
   if((status=OCIAttrGet((dvoid *)hndl,htyp,(dvoid *)attp,(dvoid
*)size,atyp,errh)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIaset(hndl,htyp,attp,size,atyp,errh) \
   if((status=OCIAttrSet((dvoid *)hndl,htyp,(dvoid
*)attp,size,atyp,errh)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIsexec(svch,stmh,errh,iter) \

if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DE
FAULT)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIbbname(stmh,bindp,errh,sqlvar,progv,progvl,ftype) \
   if((status=OCIBindByName(stmh,&bindp,errh,(text
*)sqlvar,strlen(sqlvar), \
            progv,progvl,ftype,0,0,0,0,0,OCI_DEFAULT)) !=
OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIbbnamei(stmh,bindp,errh,sqlvar,progv,progvl,ftype,indp) \
   if((status=OCIHandleAlloc((dvoid *)stmh,(dvoid
**)&bindp,OCI_HTYPE_BIND, \
                     0,(dvoid **)0))!=OCI_SUCCESS) \
      sql_error(stmh,status,0); \
   if((status=OCIBindByName(stmh,&bindp,errh,(text
*)sqlvar,strlen(sqlvar), \
            progv,progvl,ftype,indp,0,0,0,0,OCI_DEFAULT)) !=
OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIcom(svcp,errh) \
    if((status=OCITransCommit(svcp,errh,OCI_DEFAULT)) !=
OCI_SUCCESS) \
       sql_error(errh,status,1); \
    else \
       DISCARD 0

#define OCIrol(svcp,errh) \
    if((status=OCITransRollback(svcp,errh,OCI_DEFAULT)) !=
OCI_SUCCESS) \
       sql_error(errh,status,1); \
    else \
       DISCARD 0

#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree 2)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 2)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"

#define SQLTXT1 "BEGIN SELECT /*+ index(lineitem,i_l_orderkey)
*/ MAX(l_linenumber) INTO :l_key FROM lineitem \
WHERE l_orderkey = :o_key; END;"

#define SQLTXT2 "BEGIN d_atrans.doatrans(:l_key, :o_key, :delta,
:l_pkey, \
:l_skey, :l_quan, :l_newquan, :l_tax, :l_disc, :l_eprice, :l_neweprice, \
:o_tprice, :o_newtprice, :rprice, :cost); END;"

#define SQLTXT3 "BEGIN SELECT l_extendedprice, l_quantity \
INTO :l_neweprice, :l_newquan \
FROM lineitem \
WHERE l_orderkey = :o_key \
AND l_linenumber = :l_key; END;"
```

```
#define SQLTXT4 "BEGIN SELECT o_totalprice INTO :o_newtprice \
FROM orders \
WHERE o_orderkey = :o_key; END;"

#define SQLTXT5 "BEGIN SELECT l_extendedprice, l_quantity \
INTO :l_eprice, :l_quan \
FROM lineitem \
WHERE l_orderkey = :o_key \
AND l_linenumber = :l_key; END;"

#define SQLTXT6 "BEGIN SELECT o_totalprice INTO :o_tprice \
FROM orders \
WHERE o_orderkey = :o_key; END;"

#endif /* ATRANSPL_H */
```

---

ckpt.sh

```
#!/bin/ksh
#
# $Header: ckpt.sh 08-aug-99.17:37:07 mpoess Exp $
#
# ckpt.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      ckpt.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: ckpt.sh
#      Start database checkpoint
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

sqlplus -s /NOLOG << !

        connect / as sysdba;
        alter system switch logfile;
        alter system switch logfile;
        exit;
!
```

---

cnt_hist.sql

```
select count(*) from history;
exit;
```

---

consist.sh

```
#!/bin/ksh
#
# $Header: consist.sh 08-aug-99.14:20:51 mpoess Exp $
#
# consist.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      consist.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Performs consistency tests.
#      Usage: consist.sh [-n iter] [-s number of stream] [-p prog]
#                        [-u usr/pswd] -h
#
#      Options: See usage below
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit set in env
OUT_DIR=$ACID_OUT

KEY=$OUT_DIR/key$$_
OUTFILE=${OUT_DIR}/consrte
CON1=${OUT_DIR}/conb
CON2=${OUT_DIR}/cona
CHK=${OUT_DIR}/consckpt

/bin/rm -rf ${KEY}* $CON1 $CON2 $OUTFILE $CHK

trap "/bin/rm -rf ${KEY}*; exit 1" 1 2 3 15

STREAM=${NUM_STREAMS}
let STREAM="$STREAM + 1" # add one for the update stream
ITER=100
PROG=atranspl
USER=${DATABASE_USER}
CK=10

usage() {

  echo ""
  echo "Usage: $0 [-n iter] [-s number of stream] [-p prog] [-u usr/pswd]
-h"
  echo ""
  echo "-n iter            : number of iterations, default is 100"
  echo "-s number of stream : number of streams, default is 2"
  echo "-p prog            : program to run, default is atranspl.ott"
  echo "-u usr/pswd        : user/password for database access, default is
tpcd/tpcd"
  echo "-t chkpt           : time after the start of ACID transaction to
perform the checkpoint"
  echo "                     default is 10 seconds"
  echo "-h                 : print this usage summary"
  exit 1;
}

set -- `getopt "n:p:u:s:h" "$@"` || usage

while :
do
  case "$1" in
  -s) shift; STREAM=$1;;
  -n) shift; ITER=$1;;
```

```
  -p) shift; PROG=$1;;
  -u) shift; USER=$1;;
  -t) shift; CK=$1;;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift
done


if [ $ITER -lt 100 ]
then
echo "Error: Must at least run 100 iterations!"
echo "Exiting..."
exit 1
fi

if [ $STREAM -lt 2 ]
then
echo "Error: Must at least run 2 streams!"
echo "Exiting..."
exit 1
fi

echo "Starting Consistency Test at `date`..."
echo ""
echo "Generate some keys first"
echo ""

i=0

while [ $i -lt $STREAM ]
do
  echo randkey $ITER 1 u$USER
  randkey $ITER 1 u$USER > ${KEY}$i
  i=`expr $i + 1`
done

echo "Check consistency before Submitting Transactions `date`"
echo "Check consistency before Submitting Transactions `date`" >>
$CON1

echo "Obtain 10 keys from the each key file to check consistency"

i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}${i} | awk '{printf "%d ", $1}'`
echo "The 10 Keys for file $i are: $KEYS"
#for j in `head -10 ${KEY}${i} | awk '{printf "%d ", $1}'`
for j in $KEYS
do
   sqlplus $USER @consist $j >> $CON1
   echo "------------------------------" >> $CON1
done
   i=`expr $i + 1`
done

echo ""
echo "Starting ACID transactions at `date`"
echo ""

i=0

while [ $i -lt $STREAM ]
do
   $PROG $i $STREAM 1 0 u${USER} i${KEY}${i}
o${OUTFILE}${i} s1 &
   i=`expr $i + 1`
done

echo "Schedule a Checkpoint"
echo "Checkpoint scheduled at $CK seconds after `date`"

(sleep $CK; $ACID_DIR/ckpt.sh) &

wait

echo ""
echo "Ending ACID transactions at `date`"
echo ""

echo "Completed $STREAM transaction streams with $ITER iterations
each"
echo ""

echo "Check consistency after Submitting Transactions `date`"
echo "Check consistency after Submitting Transactions `date`" >>
$CON2

cat ${ORACLE_HOME}/rdbms/log/alert_${ORACLE_SID}.log >>
$CHK

i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}$i | awk '{printf "%d ", $1}'`
#for j in `head -10 ${KEY}$i | awk '{printf "%d ", $1}'`
echo "The keys to check for consistency after the test from file $i are:"
echo "$KEYS"
for j in $KEYS
do
   sqlplus $USER @consist $j >> $CON2
   echo "------------------------------" >> $CON2
done
   i=`expr $i + 1`
done
```

---

consist.sql

```
Rem
Rem $Header: consist.sql 08-aug-99.16:59:17 mpoess Exp $
Rem
Rem consist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      consist.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Verifies the consistency of TPC-D database using the
Rem       consistency condition.
Rem
Rem Usage:   sqlplus tpcd/tpcd @consist
Rem
Rem NOTE
Rem REQUIRES PACKAGES prvtotpt and dbmsotpt
rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/08/99 - Creation
Rem    mpoess      08/08/99 - Created
Rem

set verify off
rem set termout on
rem set echo on
```

```
REM
REM Get today's date.
REM

select
substr(TO_CHAR(sysdate,'YYYY-MM-DD HH:MI:SS'),1,20) as
CURRENT_TIME
from dual;

set serverout on;

DECLARE
        o_okey          number;
        o_tprice    number;
        l_tprice    number;
        diff                    number;
BEGIN
   select o_totalprice
        into o_tprice
        from orders
        where o_orderkey = &&1;


   select sum(trunc((trunc((l_extendedprice * (1-l_discount)), 2)
        * (1+l_tax)), 2))
        into l_tprice
        from lineitem
        where l_orderkey = &&1;

   diff := l_tprice - o_tprice;

   dbms_output.put_line('O_TOTALPRICE: ' ||
TO_CHAR(trunc(o_tprice,2)));
   dbms_output.put_line('L_TOTALPRICE: ' ||
TO_CHAR(trunc(l_tprice,2)));
   dbms_output.put_line('Difference:   ' || TO_CHAR(trunc(diff,2)));

END;
.
/


spool off
exit
```

---

dura.sh

```
#!/bin/ksh
#
# $Header: dura.sh 08-aug-99.15:21:38 mpoess Exp $
#
# dura.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      dura.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file declares/defines>
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

# Create history table

# Count number of entries in the history table

SERVER="ultraperf2"

echo "----------------------------------------------------"
echo "Capturing Process information before durability tests `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------------------------------------------------"

echo "----------------------------------------------------"
echo "Starting the durability tests `date`"
run_acid.sh &
echo "----------------------------------------------------"

sleep 1200

echo "----------------------------------------------------"
echo "Collecting user information. `date`"
./cnt_user.sh pswong spyda ultraperf2 > dura/duraucnt 2>&1
echo "----------------------------------------------------"

echo "----------------------------------------------------"
echo "Capturing Process information while running Transactions `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------------------------------------------------"

echo "----------------------------------------------------"
echo "Capturing disk information on Server: Ultraperf2 `date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo "----------------------------------------------------"

echo "----------------------------------------------------"
echo "Detaching mirror on data disk. `date`"
rsh $SERVER -n -l root  "vxplex -v ordr23 det ordr23-01"
echo "----------------------------------------------------"

echo "----------------------------------------------------"
echo "Capturing Disk information information on Server: Ultraperf2
`date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo "----------------------------------------------------"

sleep 120

echo "----------------------------------------------------"
echo "Capturing Process information after breaking data mirror. `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "----------------------------------------------------"


echo "----------------------------------------------------"
echo "Detaching mirror on log2 disk. `date`"
rsh $SERVER -n -l root "vxplex -v log2 det log2-01"
echo "----------------------------------------------------"

echo "----------------------------------------------------"
echo "Capturing Disk information information on Server: Ultraperf2
`date`"
rsh $SERVER -n -l spyda  vxprint -ht ; date
echo "----------------------------------------------------"

sleep 120
```

```
echo "------------------------------------------------------"
echo "Capturing Process information after detaching log mirror. `date`"
rsh $SERVER -n -l spyda  ps -ef; date
echo "------------------------------------------------------"

# Power Off
```

_____

end_acid.sh

_____

```
#!/bin/ksh
#
# $Header: end_acid.sh 08-aug-99.17:06:20 mpoess Exp $
#
# end_acid.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      end_acid.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      end_cons.sh <pid of the durability run>
#      Options: See usage below
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$OH/tpcd/audit set in env
OUT_DIR=$ACID_OUT/
DURA_DIR=$ACID_OUT/dura
RUN_ID_FILE=$ACID_DIR/run_id

SHELL_PID=`cat ${DURA_DIR}/shellpid`
ITER=100
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/drate
OUT=${DURA_DIR}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${SHELL_PID}_
USER=tpch/tpch
TRIG=1
HCNT=duracnta

# get history count

sqlplus $USER @cnt_hist > $DURA_DIR/$HCNT 2>&1

# perform the consistency

i=0
while [ $i -lt $STEM ]
do
  for j in `head -10 ${KEY}${i} | awk '{printf "%d ",$1}'`
  do
    sqlplus tpch/tpch @consist $j >> $DURA_DIR/duraconsa
  done
  i=`expr $i + 1`
done

i=0
while [ $i -lt $STEM ]
do
sample.sh $DURA${i} > ${DSMPL}${i} 2>&1
i=`expr $i + 1`
done
```

_____

gettime.c

_____

```
#ifdef RCSID
static char *RCSid =
  "$Header: gettime.c 15-jul-99.14:27:44 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) Oracle Corporation 1999. All Rights Reserved. */

/*

  NAME
    gettime.c

  DESCRIPTION
    get wall clock time.
    get cpu time.

  FUNCTIONS
    get wall clock time.
    get cpu time.

  NOTES
    Both routines return time in seconds as a double.
  MODIFIED   (MM/DD/YY)
  mpoess      07/15/99 - Creation
  mpoess      07/15/99 - Creation

*/


/*
**  Options:
**     TIME_W_TIMES:      implement gettime() with times().
**     TIME_W_GETTIME:    implement gettime() with gettimeofday().
**     CPU_W_TIMES:       implement getcpu()  with times().
**     CPU_W_GETRU:       implement getcpu()  with getrusage().
**     GETRU_STATS:       collect getrusage statistics
**     GET_P_STATS:       collect get_process_stats statistics
*/

#define SUN_OS5

#if defined(SUN_OS5)
#define TIME_W_GETTIME
#define CPU_W_TIMES
#undef GETRU_STATS
#undef CPU_W_GETRU
#endif /* SUN_OS5 */

#if defined(sequent) || defined(SEQ_PSX)
# define GET_P_STATS
#endif /* sequent */

#if defined(aix) || defined(AIXRIOS)
# define TIME_W_GETTIME
# define CPU_W_TIMES
```

```
# define GETRU_STATS
#endif /* AIXRIOS */

#if defined(a_osf) || defined(A_OSF)
# define TIME_W_GETTIME
# define CPU_W_GETRU
# define GETRU_STATS
#endif /* AIXRIOS */

#if defined(HPUX) || defined(XENIX_386) || defined(SYSV_386) ||
defined(ATT_3B)
# define TIME_W_TIMES
# define CPU_W_TIMES
#endif /* HPUX || XENIX_386 || SYSV_386 */

#if !defined(TIME_W_GETTIME) && !defined(TIME_W_TIMES)
# define TIME_W_TIMES
#endif

#if !defined(CPU_W_GETRU) && !defined(CPU_W_TIMES)
# define CPU_W_TIMES
#endif

#ifdef GET_P_STATS
# ifdef GETRU_STATS
#  undef GETRU_STATS
# endif
#endif

#if defined(TIME_W_GETTIME) || defined(CPU_W_GETRU) ||
defined(GETRU_STATS)
# include <sys/time.h>
#endif /* TIME_W_GETTIME || CPU_W_GETRU || GETRU_STATS
*/

#if defined(CPU_W_GETRU) || defined(GETRU_STATS)
# include <sys/resource.h>
#endif /* CPU_W_GETRU || GETRU_STATS */

#if defined(TIME_W_TIMES) || defined (CPU_W_TIMES)
# include <sys/types.h>
# include <sys/times.h>
# include <sys/param.h>  /* most systems define HZ here */
#endif /* TIME_W_TIMES or CPU_W_TIMES */

#ifdef GET_P_STATS
# include <sys/types.h>
# include <sys/procstats.h>
#endif /* GET_P_STATS */

# include <stdio.h>

#ifdef GETRU_STATS
struct rusage selfru;
struct rusage kidsru;
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
struct process_stats selfru;
struct process_stats kidsru;
#endif /* GET_P_STATS */



double gettime ()

{

#ifdef TIME_W_GETTIME
  struct timeval tv;

  (void) gettimeofday (&tv, (struct timezone *) 0);
  return ((double) tv.tv_sec + (1.0e-6 * (double) tv.tv_usec));
#endif /* TIME_W_GETTIME */

#ifdef TIME_W_TIMES
  struct tms buf;

  return ((double) times (&buf) / HZ);
#endif /* TIME_W_TIMES */

}



double getcpu ()

{

#ifdef CPU_W_TIMES
  struct tms buf;

  (void) times (&buf);
  return (((double) buf.tms_utime + (double) buf.tms_stime) / HZ);
#endif /* CPU_W_TIMES */

#ifdef CPU_W_GETRU
  struct rusage ru;
  double usecs;

  (void) getrusage (0, &ru);
  usecs = 1.0e-6 * (double) (ru.ru_utime.tv_usec + ru.ru_stime.tv_usec);
  return ((double) (ru.ru_utime.tv_sec + ru.ru_stime.tv_sec) + usecs);
#endif /* CPU_W_GETRU */

}



getru (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
  struct rusage ru;

  fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config,runname,
proc_no, kids);
  getrusage (kids ? RUSAGE_CHILDREN : RUSAGE_SELF, &ru);
  print_ru (fp, &ru);
  fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
  timeval_t tv;
  struct process_stats ru;

  fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config,runname,
proc_no, kids);
  if (kids)
    get_process_stats (&tv, PS_SELF, (struct process_stats *) 0, &ru);
  else
    get_process_stats (&tv, PS_SELF, &ru, (struct process_stats *) 0);
  print_ru (fp, &ru);
```

```
  fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



getru1 (kids)

int kids;

{

#ifdef GETRU_STATS
  if (kids) {
    memset (&kidsru, 0, sizeof (kidsru));
    getrusage (RUSAGE_CHILDREN, &kidsru);
  }
  else {
    memset (&selfru, 0, sizeof (selfru));
    getrusage (RUSAGE_SELF, &selfru);
  }
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
  timeval_t tv;

  if (kids) {
    memset (&kidsru, 0, sizeof (kidsru));
    get_process_stats (&tv, PS_SELF, (struct process_stats *) 0,
&kidsru);
  }
  else {
    memset (&selfru, 0, sizeof (selfru));
    get_process_stats (&tv, PS_SELF, &selfru, (struct process_stats *)
0);
  }
#endif /* GET_P_STATS */

}



getru2 (fp, kids, config, runname, proc_no)

FILE *fp;
int kids;
char *config;
char *runname;
int proc_no;

{

#ifdef GETRU_STATS
  struct rusage ru;

  fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname,
proc_no, kids);
  getrusage (kids ? RUSAGE_CHILDREN : RUSAGE_SELF, &ru);
  if (kids)
    diffru (&ru, &kidsru);
  else
    diffru (&ru, &selfru);
  print_ru (fp, &ru);
  fprintf (fp, "\n");
#endif /* GETRU_STATS */

#ifdef GET_P_STATS
  timeval_t tv;
  struct process_stats ru;

  fprintf (fp, "%-10.10s %-10.10s %10d %10d ", config, runname,
proc_no, kids);
  if (kids)
    get_process_stats (&tv, PS_SELF, (struct process_stats *) 0, &ru);
  else
    get_process_stats (&tv, PS_SELF, &ru, (struct process_stats *) 0);
  if (kids)
    diffru (&ru, &kidsru);
  else
    diffru (&ru, &selfru);
  print_ru (fp, &ru);
  fprintf (fp, "\n");
#endif /* GET_P_STATS */

}



#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

  fprintf (fp, "%10ld ", ru->ru_utime.tv_sec * 1000 +
                  (ru->ru_utime.tv_usec/1000));
  fprintf (fp, "%10ld ", ru->ru_stime.tv_sec * 1000 +
                  (ru->ru_stime.tv_usec/1000));
  fprintf (fp, "%10ld ", ru->ru_maxrss);
  fprintf (fp, "%10ld ", ru->ru_majflt);
  fprintf (fp, "%10ld ", ru->ru_minflt);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld ", ru->ru_nswap);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld ", ru->ru_nvcsw);
  fprintf (fp, "%10ld ", ru->ru_nivcsw);
  fprintf (fp, "%10ld ", ru->ru_nsignals);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld ", ru->ru_inblock);
  fprintf (fp, "%10ld ", ru->ru_oublock);
  fprintf (fp, "%10ld ", 0);
  fprintf (fp, "%10ld",  0);

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

  ru2->ru_utime.tv_sec -= ru->ru_utime.tv_sec;
  ru2->ru_utime.tv_usec -= ru->ru_utime.tv_usec;
  ru2->ru_stime.tv_sec -= ru->ru_stime.tv_sec;
  ru2->ru_stime.tv_usec -= ru->ru_stime.tv_usec;
  ru2->ru_maxrss -= ru->ru_maxrss;
  ru2->ru_ixrss -= ru->ru_ixrss;
  ru2->ru_idrss -= ru->ru_idrss;
  ru2->ru_minflt -= ru->ru_minflt;
  ru2->ru_majflt -= ru->ru_majflt;
```

```
  ru2->ru_nswap -= ru->ru_nswap;
  ru2->ru_inblock -= ru->ru_inblock;
  ru2->ru_oublock -= ru->ru_oublock;
  ru2->ru_msgsnd -= ru->ru_msgsnd;
  ru2->ru_msgrcv -= ru->ru_msgrcv;
  ru2->ru_nsignals -= ru->ru_nsignals;
  ru2->ru_nvcsw -= ru->ru_nvcsw;
  ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

  fprintf (fp, "%lu ", ps->ps_utime.tv_sec * 1000 +
                  (ps->ps_utime.tv_usec/1000));
  fprintf (fp, "%lu ", ps->ps_stime.tv_sec * 1000 +
                   (ps->ps_stime.tv_usec/1000));
  fprintf (fp, "%lu ", ps->ps_maxrss);
  fprintf (fp, "%lu ", ps->ps_pagein);
  fprintf (fp, "%lu ", ps->ps_reclaim);
  fprintf (fp, "%lu ", ps->ps_zerofill);
  fprintf (fp, "%lu ", ps->ps_pffincr);
  fprintf (fp, "%lu ", ps->ps_pffdecr);
  fprintf (fp, "%lu ", ps->ps_swap);
  fprintf (fp, "%lu ", ps->ps_syscall);
  fprintf (fp, "%lu ", ps->ps_volcsw);
  fprintf (fp, "%lu ", ps->ps_involcsw);
  fprintf (fp, "%lu ", ps->ps_signal);
  fprintf (fp, "%lu ", ps->ps_lread);
  fprintf (fp, "%lu ", ps->ps_lwrite);
  fprintf (fp, "%lu ", ps->ps_bread);
  fprintf (fp, "%lu ", ps->ps_bwrite);
  fprintf (fp, "%lu ", ps->ps_phread);
  fprintf (fp, "%lu",  ps->ps_phwrite);

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

  ru2->ps_utime.tv_sec -= ru->ps_utime.tv_sec;
  ru2->ps_utime.tv_usec -= ru->ps_utime.tv_usec;
  ru2->ps_stime.tv_sec -= ru->ps_stime.tv_sec;
  ru2->ps_stime.tv_usec -= ru->ps_stime.tv_usec;
  ru2->ps_maxrss -= ru->ps_maxrss;
  ru2->ps_pagein -= ru->ps_pagein;
  ru2->ps_reclaim -= ru->ps_reclaim;
  ru2->ps_zerofill -= ru->ps_zerofill;
  ru2->ps_pffincr -= ru->ps_pffincr;
  ru2->ps_pffdecr -= ru->ps_pffdecr;
  ru2->ps_swap -= ru->ps_swap;
  ru2->ps_syscall -= ru->ps_syscall;
  ru2->ps_volcsw -= ru->ps_volcsw;
  ru2->ps_involcsw -= ru->ps_involcsw;
  ru2->ps_signal -= ru->ps_signal;
  ru2->ps_lread -= ru->ps_lread;
  ru2->ps_lwrite -= ru->ps_lwrite;
  ru2->ps_bread -= ru->ps_bread;
  ru2->ps_bwrite -= ru->ps_bwrite;
  ru2->ps_phread -= ru->ps_phread;
  ru2->ps_phwrite -= ru->ps_phwrite;

}

#endif /* GET_P_STATS */
```

_____

gtime.c

_____

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*

  NAME
    gtime.c - <one-line expansion of the name>

  DESCRIPTION
    <short description of facility this file declares/defines>

  EXPORT FUNCTION(S)
    <external functions defined for use outside package - one-line
descriptions>

  INTERNAL FUNCTION(S)
    <other external functions defined - one-line descriptions>

  STATIC FUNCTION(S)
    <static functions defined - one-line descriptions>

  NOTES
    <other useful comments, qualifications, etc.>

  MODIFIED   (MM/DD/YY)
  mpoess      10/23/02 - mpoess_update_from_visa
  mpoess      08/29/01 - Creation

*/
#include<stdio.h>
#include<stdlib.h>

# include <sys/time.h>

main ()
{

 struct timeval tv;

      (void) gettimeofday (&tv, (struct timezone *) 0);

 printf ("%.2f\n", ((double) tv.tv_sec + (1.0e-6 * (double) tv.tv_usec)) )
;

}

/* end of file gtime.c */
```

_____

iso1.sh

_____

```
#!/bin/ksh
```

```
#
# $Header: iso1.sh 29-jul-98.17:00:11 akarasik Exp $
#
# iso1.sh
#
#  Copyright (c) Oracle Corporation 1998. All Rights Reserved.
#
#    NAME
#      iso1.sh
#
#  DESCRIPTION
#       Usage: iso1.sh [-u user/password] [-n remote_node] -h
#       Options: See usage below
#  NOTES
#               For a cross node isolation test, assume the local node is
#               one of the participating nodes.  The other node can be
#               specified by the -n option.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      12/16/98 - update to version 8.1.6
#    mpoess      09/25/98 - update audit
#    akarasik    07/29/98 -
#    akarasik    07/29/98 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
#ACID_DIR=$KIT_DIR/acid is set in env
OUT_DIR=$ACID_OUT

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso1

USER=$DATABASE_USER
PROG=atranspl


/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

  echo ""
  echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
  echo ""
  exit 1;
}

set -- `getopt "u:n:h" "$@"` || usage

while :
do
  case "$1" in
  -u) shift; USER=$1;;
  -n) shift; HOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift;
done

de=`direxists.sh $ACID_OUT c` # I am not using $de afterward, but I
want to avoid the output of direxists

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the ACID transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 1" >>
$TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
echo "" >> $TXN2FILE
echo "------------------------------------------------" >> $TXN2FILE

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

$PROG 1 1 1 0 i$KEYFILE u$USER s60 b0 >> $TXN1FILE &

# let's sleep 10 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

echo "Running ACID query 15 seconds AFTER the start of ACID
Transaction" \
>> $TXN2FILE
echo ${RSH} -n ${HOST}  sqlplus $USER
@$ACID_DIR/isolation/a_query $OKEY
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
echo ${RSH} -n ${HOST}  sqlplus $USER
@$ACID_DIR/isolation/a_query $OKEY
${RSH} -n ${HOST}  sqlplus $USER @$ACID_DIR/isolation/a_query
$OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

echo "------------------------------------------------" >> $TXN2FILE
wait
echo "------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

---

iso2.sh

```
#!/bin/ksh
#
# $Header: iso2.sh 04-aug-99.09:19:54 mpoess Exp $
#
# iso2.sh
```

```
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso2.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      Usage: iso2.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#    NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#
#
#
================================================
===============+
# May need to change the following:

. $KIT_DIR/env

RSH=rsh

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso2

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

  echo ""
  echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
  echo ""
  exit 1;
}

set -- `getopt "u:n:h" "$@"` || usage

while :
do
  case "$1" in
  -u) shift; USER=$1;;
  -n) shift; HOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift;
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the ACID transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 1" >>
$TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus "$USER" @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
echo "" >> $TXN2FILE
echo "-------------------------------------------------" >> $TXN2FILE

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

$PROG 1 1 0 0 i$KEYFILE u$USER s30 >> $TXN1FILE &

# let's sleep 15 seconds before starting ACID query

sleep 15

# start ACID query with the same OKEY

echo "Running ACID query 15 seconds AFTER the start of ACID
transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST}  sqlplus "$USER"
@$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

echo "-------------------------------------------------" >> $TXN2FILE
wait
echo "-------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

_____

iso3.sh

_____

```
#!/bin/ksh
#
# $Header: iso3.sh 04-aug-99.09:20:35 mpoess Exp $
#
# iso3.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso3.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso3.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
```

```
#  NOTES
#      For a cross node isolation test, assume the local node is
#      one of the participating nodes.  The other node can be
#      specified by the -n option.
#            We need to make sure the remote node has access to the
#            file system on the local node.  Otherwise, we need to rcp
#            the keyfile to the remote system.
#      You need to set the environment variable TPCD_KIT_DIR
#
#   MODIFIED   (MM/DD/YY)
#   mpoess      08/04/99 - Creation
#   mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
#ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso3

USER=$DATABASE_USER
PROG=$KIT_DIR/utils/atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

  echo ""
  echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
  echo ""
  exit 1;
}

set -- `getopt "u:n:h" "$@"` || usage

while :
do
  case "$1" in
  -u) shift; USER=$1;;
  -n) shift; HOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
echo Host $HOST
rcp $KEYFILE ${HOST}:$KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

$PROG 1 2 1 0 i$KEYFILE u$USER s60 b0 >> $TXN1FILE &

# let's sleep 15 seconds before starting second ACID transaction

sleep 15

# start another ACID transaction with the same LKEY and OKEY
# but different DELTA

# Do not sleep before COMMIT so that we can see TXN2 has waited.

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >>
$TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

wait
echo "------------------------------------------------" >> $TXN2FILE
echo "------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

iso4.sh

```
#!/bin/ksh
#
# $Header: iso4.sh 04-aug-99.09:21:12 mpoess Exp $
#
# iso4.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso4.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso4.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#  NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       We need to make sure the remote node has access to the
#       file system on the local node.  Otherwise, we need to rcp
#       the keyfile to the remote system.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura
```

```
TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso4

USER=$DATABASE_USER
PROG=$KIT_DIR/utils/atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

  echo ""
  echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
  echo ""
  exit 1;
}

set -- `getopt "u:n:h" "$@"` || usage

while :
do
  case "$1" in
  -u) shift; USER=$1;;
  -n) shift; HOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

$PROG 1 2 0 0 i$KEYFILE u$USER s30 b0 >> $TXN1FILE &

# let's sleep 15 seconds before starting second ACID transaction

sleep 15

# start another ACID transaction with the same LKEY and OKEY
# but different DELTA

# Do not sleep before COMMIT so that we can see TXN2 has waited.

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >>
$TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 b1 >> $TXN2FILE &
fi

wait
echo "----------------------------------------------------" >> $TXN2FILE
echo "----------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

iso5.sh

```
#!/bin/ksh
#
# $Header: iso5.sh 04-aug-99.09:21:45 mpoess Exp $
#
# iso5.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso5.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso5.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#  NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso5

USER=$DATABASE_USER
PROG=atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE; exit 1" 1 2 3 15

usage() {

  echo ""
  echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
  echo ""
  exit 1;
}

set -- `getopt "u:n:h" "$@"` || usage

while :
do
  case "$1" in
  -u) shift; USER=$1;;
  -n) shift; HOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
```

```
  shift;
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the ACID transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 5" >>
$TXN1FILE
echo "`date`" >> $TXN1FILE
echo "" >> $TXN1FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN1FILE
echo "" >> $TXN1FILE
echo "-------------------------------------------------" >> $TXN1FILE

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

$PROG 1 1 1 0 i$KEYFILE u$USER s60 >> $TXN1FILE &

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY

PSKEY=`randpsup 1`

echo "Running PARTSUPP query 5 seconds AFTER the start of ACID
Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
echo "PS_PARTKEY and PS_SUPPKEY are: $PSKEY" >>
$TXN2FILE

if [ "$HOST" != "" ]
then
echo "Starting PARTSUPP query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} sqlplus $USER
@$ACID_DIR/isolation/a_query2 ${PSKEY} >> $TXN2FILE &
else
sqlplus $USER @$ACID_DIR/isolation/a_query2 ${PSKEY} >>
$TXN2FILE &
fi

wait

echo "-------------------------------------------------" >> $TXN2FILE
echo "-------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $KEYFILE
```

_____

iso6.sh

_____

```
#!/bin/ksh
# $Header: iso6.sh 04-aug-99.09:22:12 mpoess Exp $
#
# iso6.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso6.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso6.sh [-u user/password] [-n remote_node] -h
#      Options: See usage below
#  NOTES
#       For a cross node isolation test, assume the local node is
#       one of the participating nodes.  The other node can be
#       specified by the -n option.
#       We need to make sure the remote node has access to the
#       file system on the local node.  Otherwise, we need to rcp
#       the keyfile to the remote system.
#       You need to set the environment variable TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

OH=/private/tpcd
# ACID_DIR=$TPCD_KIT_DIR/audit is set in env
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
TXN3FILE=$OUT_DIR/txn3$$.out
KEYFILE=$OUT_DIR/key$$.out
ISOFILE=$OUT_DIR/iso6

USER=$DATABASE_USER
PROG=$KIT_DIR/utils/atranspl

/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE

trap "/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE;
exit 1" 1 2 3 15

usage() {

  echo ""
  echo "Usage: $0 [-u user/passwd] [-n remote_node] -h"
  echo ""
  exit 1;
}

set -- `getopt "u:n:h" "$@"` || usage

while :
do
  case "$1" in
  -u) shift; USER=$1;;
  -n) shift; HOST="$1";;
  -h) usage; exit 0;;
  --) break;;
  esac
  shift;
```

```
done


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE
rcp $KEYFILE ${HOST}:$KEYFILE

OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

# before the any transaction, let's run a ACID query to record the
# initial state of lineitem

echo "Running ACID query BEFORE the start of Isolation Test 6" >>
$TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
echo "" >> $TXN2FILE
echo "-------------------------------------------------" >> $TXN2FILE

sleep 1

# start Query 1, use 0 as the delta

echo "Running Query 17b at `date`" >> $TXN1FILE
sqlplus $USER @q1 >> $TXN1FILE &

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

echo "Starting AICD transaction at `date`" >> $TXN2FILE

if [ "$HOST" != "" ]
then
echo "Starting ACID transaction on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} $PROG 1 1 1 0 i$KEYFILE u$USER s1 >>
$TXN2FILE &
else
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

# start Query 17

sleep 2

echo "Running 2nd Query 17b at `date`" >> $TXN3FILE
sqlplus $USER @q1 >> $TXN3FILE &
# wait for everyone to finish

wait

echo "-------------------------------------------------" >> $TXN3FILE
echo "-------------------------------------------------" >> $TXN2FILE
echo "-------------------------------------------------" >> $TXN1FILE

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE
```

randkey.c

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*

  NAME
    randkey.c - <one-line expansion of the name>

  DESCRIPTION
    Generate random keys for ACID transactions:
    O_ORDERKEY unique random (1..SF*150000*4) and only
    first 8 keys out of every 32 are populated.
      and
    L_ORDERKEY based on Clause 3.1.6.2
    DELTA random (1..100)
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "atranspl.h"

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

#define  MK_SPARSE(key, seq) \
        (((((key>>3)<<2)|(seq & 0x0003))<<3)|(key & 0x0007))

void sql_error();
void usage();
void ACIDinit();
long atol();
void srand48();
long lrand48();

/* Not really used here, but retained it for future purposes. */

typedef struct aciddef {
  long okey;
  long lkey;
  int delta;
} adef;

long l_key = 0;
long o_key = 0;
char lname[UNAME_LEN];
char *passwd;

/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

sword status = OCI_SUCCESS;  /* OCI return value */

char sqlstmt[1024];


void ACIDexit() {
  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);
```

```
}


/* type: 0 if environment handle is passed, 1 if error handle is passwd */

void sql_error(errhp,status,type)
    OCIError *errhp;
    sword status;
    sword type;
{
 char msg[2048];
 sb4 errcode;
 ub4 msglen;
 int i,j;

 switch(status) {
 case OCI_SUCCESS_WITH_INFO:
   fprintf(stderr, "Error: Statement returned with info.\n");
   if (type)
     (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                  2048,OCI_HTYPE_ERROR);
   else
     (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                  2048,OCI_HTYPE_ENV);
   fprintf(stderr,"%s\n",msg);
   break;
 case OCI_ERROR:
   fprintf(stderr, "Error: OCI call error.\n");
   if (type)
     (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                  2048,OCI_HTYPE_ERROR);
   else
     (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                  2048,OCI_HTYPE_ENV);
   fprintf(stderr,"%s\n",msg);
   break;
 case OCI_INVALID_HANDLE:
   fprintf(stderr, "Error: Invalid Handle.\n");
   if (type)
     (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                  2048,OCI_HTYPE_ERROR);
   else
     (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text *)msg,
                  2048,OCI_HTYPE_ENV);
   fprintf(stderr,"%s\n",msg);
   break;
 }
 /* Rollback just in case */

 (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

 fprintf(stderr, "Exiting Oracle...\n");
 fflush(stderr);

 ACIDexit();

 exit(1);
}


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

 long count;
 long i;
 double sf;        /* need to accomodate sf 0.1 */
 double random;
 double ordcnt;
 adef *res;

 if ((argc < 3) || (argc > 4)) {
   usage();
   exit(-1);
 }

 strcpy((char *) lname, "tpcd/tpcd");

 count = atol(argv[1]);
 sf = atof(argv[2]);

 argc -= 2;
 argv += 2;

 while (--argc) {
   ++argv;
   switch(argv[0][0]) {
   case 'u':
     strncpy((char *) lname, ++(argv[0]), UNAME_LEN);
     if (strchr((char *) lname, '/') == NULL) {
       usage();
       exit(-1);
     }
     break;
   default:
     fprintf(stderr, "Unknown argument %s\n", argv[0]);
     usage();
     break;
   }
 }

 ACIDinit();

 /* initialize array for random numbers */

 res = (adef *) malloc(count*sizeof(adef));
 ordcnt = (double) ORDERCNT * (double) sf;

 for (i=0; i<count; i++) {

   /* The algorithm:                                          */
   /* Assumes drand's output is 'unique', first get a number within */
   /* the range of [0..sf*ORDERCNT) and then maps the different     */
   /* ranges to generate the real output.                     */

   random = floor(drand48() * (double) ordcnt) + 1;
   res[i].okey = o_key = (long) MK_SPARSE((long) random, 0);
   res[i].delta = (long) floor(drand48() * 100) + 1;

   /* Obtain l_key from l_key query */

   OCIsexec(tpcsvc,curi,errhp,1);

   /* l_key is the highest l_linenumber available.  We need to pick */
   /* at random a number between 1..l_key.                    */

   res[i].lkey = (lrand48() % l_key) + 1;

   printf("%ld %ld %d\n", res[i].okey, res[i].lkey, res[i].delta);
 }

 ACIDexit();
 free(res);

}


void usage() {
```

```
 fprintf(stderr, "Usage: randkey <number of random keys to generate>
<SF> u<user/password>\n");
 fprintf(stderr, "\n");
}

void ACIDinit()
{

 /* run random seed */

 srand48(getpid());

 /* Connect to ORACLE.  Program will call sql_error()
    if an error occurs in connecting to the default database. */

 (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);
 if((status=OCIEnvInit((OCIEnv **)&tpcenv,OCI_DEFAULT,0,(dvoid
**)0)) !=
    OCI_SUCCESS)
   sql_error(tpcenv, status, 0);

 OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
 OCIhalloc(tpcenv,&curi,OCI_HTYPE_STMT);
 OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
 OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
 OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

 /* get username and password */

 passwd = strchr(lname, '/');
 *passwd = '\0';
 passwd++;

 if ((status=OCIServerAttach(tpcsrv,errhp,(text
*)0,0,OCI_DEFAULT))!=OCI_SUCCESS)
   sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER
,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_ATT
R_USERNAME,
        errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_A
TTR_PASSWORD,
        errhp);

 if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                        OCI_DEFAULT)) != OCI_SUCCESS)
   sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESSIO
N,errhp);

 /* Open and Parse cursor for query to choose determine l_key. */
 /* Binds l_key to :l_key.                                     */

 sprintf((char *) sqlstmt,SQLTXT1);
 OCIStmtPrepare(curi,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
             OCI_NTV_SYNTAX,OCI_DEFAULT);


OCIbbname(curi,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),SQL
T_INT);


OCIbbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQ
LT_INT);
}
```

randpsup.c

```
/* Copyright (c) 2001, 2002, Oracle Corporation.  All rights reserved.  */

/*

  NAME
    randpsup.c - <one-line expansion of the name>

  DESCRIPTION
    Generate random keys for ACID PARTSUPP transactions:
    (Clause 4.2.3)
    PS_PARTKEY random within [SF*200000]
      and
    PS_SUPPKEY = (PS_PARTKEY + (i * ((S/4) +
(int)(PS_PARTKEY - 1)
                /S))) % S + 1
    where i random within [0..3] and S = SF * 10000

MODIFIED
  mpoess      10/23/02 - mpoess_update_from_visa
  mpoess      01/04/01 - Creation

*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PS_PER_SF 200000.0
#define S_PER_SF 10000.0
#define SUPP_PER_PART 4

/* borrowed from build.c in the dbgen distribution */

#define PART_SUPP_BRIDGE(tgt, p, s) \
   { \
   long tot_scnt = (long) (S_PER_SF * sf); \
   tgt = (p + s *  (tot_scnt / SUPP_PER_PART +  \
      (long) ((p - 1) / tot_scnt))) % tot_scnt + 1; \
   }

void usage();
double atof();
void srand48();
long lrand48();


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

 double sf = 0.1;          /* scale factor      */
 long supp;               /* the i-th supplier */
 long pkey;               /* partkey           */
 long maxpkey;             /* highest partkey   */
 long ps_skey;             /* ps_suppkey        */

 if (argc < 2) {
  usage();
  exit(-1);
 }
```

```
 /* seed the random number generator */

 srand48(getpid());

 sf = atof(argv[1]);
 maxpkey = (long) (sf * PS_PER_SF);
 supp = lrand48() % 4;
 pkey = lrand48() % maxpkey + 1;

 PART_SUPP_BRIDGE(ps_skey, pkey, supp);

 fprintf(stdout, "%ld %ld", pkey, ps_skey);

 exit(0);
}


void usage()
{
 fprintf(stderr, "Usage: randpsup <SF>\n\n");
}
```

_____

run_acid.sh

_____

```
#!/bin/ksh
#
# $Header: run_acid.sh 08-aug-99.15:30:10 mpoess Exp $
#
# run_acid.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      run_acid.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: run_acid.sh [-n iter] [-s stream] [-p prog] [-i infile]
#                         [-o outfile] [-d durafile] [-u usr/pswd]
#                         [-t trigger] [-f scale factor] -h
#
#      Options: See usage below
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
ACID_DIR=$ACID_DIR
OUT_DIR=$ACID_OUT

usage() {

   echo ""
   echo "Usage: $0 [-n iter] [-s stream] [-p prog] [-i infile] [-o outfile]"
   echo "          [-d durafile] [-u usr/pswd] -h"
   echo ""
   echo "-n iter     : number of iterations, default is 100"
   echo "-s stream   : number of streams, default is 2"
   echo "-p prog     : program to run, default is atranspl.ott"
   echo "-i infile   : input file prefix, suffix by process number within a"
   echo "              stream and run ID, default is ./acid_in"
   echo "-o outfile  : output file prefix, similar to input file"
   echo "              default is ./out/acid_out"
   echo "-d durafile : durability file prefix, used for durability tests"
   echo "              default is ./dura/acid_dura"
   echo "-u usr/pswd : user/password combo for database access, default
is tpch/tpch"
   echo "-t trigger  : trigger time between process starts, default is 1
second"
   echo "-h          : print this usage summary"
   exit 1;
}

ITER=600
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/drate
DURA=$DURA_DIR/dura
KEY=${DURA_DIR}/key$$_
echo "$$" > ${DURA_DIR}/shellpid
USER=tpch/tpch
TRIG=1
HCNT=duracntb

set -- `getopt "n:s:p:i:o:d:u:ht:f:" "$@"` || usage

# get all the options

while :
do
   case "$1" in
   -n) shift; ITER=$1;;
   -s) shift; STEM=$1;;
   -p) shift; PROG=$1;;
   -i) shift; IN=$1;;
   -o) shift; OUT=$1;;
   -d) shift; DURA=$1;;
   -u) shift; USER=$1;;
   -h) usage; exit 0;;
   -t) shift; TRIG=$1;;
   -f) shift; SF=$1;;
   --) break;;
   esac
   shift;
done


echo "Starting ACID run..."

i=0
T=`expr $STEM \* $TRIG + 6`

# Get history count before the run

sqlplus $USER @cnt_hist > $DURA_DIR/$HCNT 2>&1

while [ $i -lt $STEM ]
do
   randkey $ITER ${SF} u${USER} > ${KEY}${i} &
   i=`expr $i + 1`
done

wait
# perform the consistency

i=0
while [ $i -lt $STEM ]
do
```

```
  for j in `head -10 ${KEY}${i} | awk '{printf "%d ",$1}'`
  do
    sqlplus tpch/tpch @consist $j >> $DURA_DIR/duraconsb
  done
  i=`expr $i + 1`
done

echo "Starting Transaction Counting Program"
count_tx.sh $STEM 100 $DURA_DIR &

i=0
while [ $i -lt $STEM ]
do

  $PROG $i $STEM 1 0 i${KEY}${i} o${OUT}${i} d${DURA}${i}
u$USER s1 &
  T=`expr $T - $TRIG`
  i=`expr $i + 1`

done

wait

echo "ACID run completed"
```

sample.sh

```
#!/bin/ksh
#
# $Header: sample.sh 08-aug-99.17:10:00 mpoess Exp $
#
# sample.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      sample.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file declares/defines>
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

# $1 durability output file

. $KIT_DIR/env

cat $1 | grep o_key | awk '{printf "%d \n", $2}' | head -106 >
/tmp/okey$$
cat $1 | grep l_key | awk '{printf "%d \n", $2}' | head -106 > /tmp/lkey$$



paste /tmp/okey$$ /tmp/lkey$$ > /tmp/keys$$
tail -6 /tmp/keys$$ > /tmp/6keys$$

echo "Keys chosen are:"
cat /tmp/6keys$$

i=1
while [ $i -le 6 ]
do

j=`cat /tmp/6keys$$ | tail -${i} | head -1`
sqlplus tpch/tpch @sample $j
i=`expr $i + 1`
done

#/bin/rm -f /tmp/*key*
```

sample.sql

```
Rem
Rem $Header: sample.sql 08-aug-99.17:10:34 mpoess Exp $
Rem
Rem sample.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      sample.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem      <short description of component this file declares/defines>
Rem
Rem    NOTES
Rem      <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/08/99 - Creation
Rem    mpoess      08/08/99 - Created
Rem

alter session set nls_date_format = 'YYYY-MM-DD HH:MI:SS';
select * from history where  h_o_key = &&1 and h_l_key = &&2;

exit;
```

Appendix D: Query text and output

_____
1.log
_____

Begin Execution at Fri Jan 23 19:38:54 2004

-- using default substitutions

```
select
l_returnflag,
l_linestatus,
sum(l_quantity) as sum_qty,
sum(l_extendedprice) as sum_base_price,
sum(l_extendedprice * (1 - l_discount)) as sum_disc_price,
sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) as sum_charge,
avg(l_quantity) as avg_qty,
avg(l_extendedprice) as avg_price,
avg(l_discount) as avg_disc,
count(*) as count_order
from
lineitem
where
l_shipdate <= to_date ('1998-12-01','YYYY-MM-DD')  -  90
group by
l_returnflag,
l_linestatus
order by
l_returnflag,
l_linestatus
```

```
L_RETURNFLAG L_LINESTATUS SUM_QTY
SUM_BASE_PRICE
SUM_DISC_PRICE        SUM_CHARGE            AVG_QTY
AVG_PRICE             AVG_DISC              COUNT_ORDER
A            F            37734107.00           56586554400.73
53758257134.87        55909065222.83        25.52
38273.13              0.05                  1478493.00
N            F            991417.00             1487504710.38
1413082168.05         1469649223.19         25.52
38284.47              0.05                  38854.00
N            O            74476040.00           111701729697.74
106118230307.61       110367043872.50       25.50
38249.12              0.05                  2920374.00
R            F            37719753.00           56568041380.90
53741292684.60        55889619119.83        25.51
38250.85              0.05                  1478870.00
```

4 rows processed.
Query Processed in 7.06 seconds.

Ended Executing this Stream at Fri Jan 23 19:39:01 2004

Stream Started at 1074908334.76
Stream Ended at 1074908341.81
Stream Processed in 7.06 seconds

SQL statements processed: 1

_____
2.log
_____

Begin Execution at Fri Jan 23 19:39:01 2004

-- using default substitutions

```
select * from (
select
s_acctbal,
s_name,
n_name,
p_partkey,
p_mfgr,
s_address,
s_phone,
s_comment
from
part,
supplier,
partsupp,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and p_size = 15
and p_type like '%BRASS'
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'EUROPE'
and ps_supplycost = (
select
min(ps_supplycost)
from
partsupp,
supplier,
nation,
region
where
p_partkey = ps_partkey
and s_suppkey = ps_suppkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'EUROPE'
)
order by
s_acctbal desc,
n_name,
s_name,
p_partkey
)
where rownum <= 100
```

```
S_ACCTBAL          S_NAME                N_NAME
P_PARTKEY          P_MFGR
S_ADDRESS                        S_PHONE
S_COMMENT
9938.53            Supplier#000005359    UNITED KINGDOM
185358.00          Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04              33-429-790-6131
blithely silent pinto beans are furiously. slyly final deposits acros
9937.84            Supplier#000005969    ROMANIA
108438.00          Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa     29-520-692-3537
carefully slow deposits use furiously. slyly ironic platelets above the
ironic
9936.22            Supplier#000005250    UNITED KINGDOM



<.....lines deleted .....>



8363.00            Manufacturer#4
WCNfBPZeSXh3h,c                       32-454-883-3821
blithely regular deposits
7850.66            Supplier#000001518    UNITED KINGDOM
86501.00           Manufacturer#1
ONda3YJiHKJOC                         33-730-383-3892
furiously final accounts wake carefully idle requests. even dolphins
wake acc
7843.52            Supplier#000006683    FRANCE
11680.00           Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy         16-464-517-8943
carefully bold accounts doub


100 rows processed.
Query Processed in 0.49 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:02 2004


Stream Started at 1074908341.89
Stream Ended at 1074908342.39
Stream Processed in 0.49 seconds


SQL statements processed: 1


_____________________________________
          3.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:02 2004



-- using default substitutions


select * from (
select
l_orderkey,
sum(l_extendedprice * (1 - l_discount)) as revenue,
o_orderdate,
o_shippriority
from
customer,
orders,
lineitem
where
c_mktsegment = 'BUILDING'
and c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate < to_date( '1995-03-15', 'YYYY-MM-DD')
and l_shipdate > to_date( '1995-03-15', 'YYYY-MM-DD')
group by
l_orderkey,
o_orderdate,
o_shippriority
order by
revenue desc,
o_orderdate)
where rownum <= 10

L_ORDERKEY         REVENUE            O_ORDERDATE
O_SHIPPRIORITY
2456423.00         406181.01          1995-03-05  0.00
3459808.00         405838.70          1995-03-04  0.00
492164.00          390324.06          1995-02-19  0.00
1188320.00         384537.94          1995-03-09  0.00
2435712.00         378673.06          1995-02-26  0.00
4878020.00         378376.80          1995-03-12  0.00
5521732.00         375153.92          1995-03-13  0.00
2628192.00         373133.31          1995-02-22  0.00
993600.00          371407.46          1995-03-05  0.00
2300070.00         367371.15          1995-03-13  0.00


10 rows processed.
Query Processed in 3.60 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:06 2004


Stream Started at 1074908342.47
Stream Ended at 1074908346.07
Stream Processed in 3.60 seconds


SQL statements processed: 1


_____________________________________
          4.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:06 2004



-- using default substitutions


select
o_orderpriority,
count(*) as order_count
from
orders
where
o_orderdate >= to_date( '1993-07-01',  'YYYY-MM-DD')
and o_orderdate < add_months(to_date( '1993-07-01', 'YYYY-MM-
DD'),3)
and exists (
select
*
from
lineitem
```

```
where
l_orderkey = o_orderkey
and l_commitdate < l_receiptdate
)
group by
o_orderpriority
order by
o_orderpriority

O_ORDERPRIORITY ORDER_COUNT
1-URGENT       10594.00
2-HIGH         10476.00
3-MEDIUM       10410.00
4-NOT SPECIFIED 10556.00
5-LOW          10487.00


5 rows processed.
Query Processed in 4.10 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:10 2004


Stream Started at 1074908346.14
Stream Ended at 1074908350.24
Stream Processed in 4.10 seconds


SQL statements processed: 1


_____________________________________
          5.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:10 2004



-- using default substitutions


select
n_name,
sum(l_extendedprice * (1 - l_discount)) as revenue
from
customer,
orders,
lineitem,
supplier,
nation,
region
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and l_suppkey = s_suppkey
and c_nationkey = s_nationkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'ASIA'
and o_orderdate >= to_date( '1994-01-01',  'YYYY-MM-DD')
and o_orderdate < add_months(to_date( '1994-01-01',  'YYYY-MM-
DD'),  12)
group by
n_name
order by
revenue desc

N_NAME                REVENUE
INDONESIA              55502041.17
VIETNAM                55295087.00
CHINA                  53724494.26
INDIA                  52035512.00
JAPAN                  45410175.70


5 rows processed.
Query Processed in 7.31 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:17 2004


Stream Started at 1074908350.32
Stream Ended at 1074908357.63
Stream Processed in 7.31 seconds


SQL statements processed: 1


_____________________________________
          6.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:17 2004



-- using default substitutions


select
sum(l_extendedprice * l_discount) as revenue
from
lineitem
where
l_shipdate >= to_date( '1994-01-01',  'YYYY-MM-DD')
and l_shipdate < add_months(to_date( '1994-01-01',  'YYYY-MM-DD'),
12)
and l_discount between .06 - 0.01 and .06 + 0.01
and l_quantity < 24

REVENUE
123141078.23


1 row processed.
Query Processed in 1.10 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:18 2004


Stream Started at 1074908357.70
Stream Ended at 1074908358.80
Stream Processed in 1.10 seconds


SQL statements processed: 1


_____________________________________
          7.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:18 2004



-- using default substitutions
```

```
select
supp_nation,
cust_nation,
l_year,
sum(volume) as revenue
from
(
select
n1.n_name as supp_nation,
n2.n_name as cust_nation,
                                    to_number (to_char
(l_shipdate,'yyyy')) as l_year,
l_extendedprice * (1 - l_discount) as volume
from
supplier,
lineitem,
orders,
customer,
nation n1,
nation n2
where
s_suppkey = l_suppkey
and o_orderkey = l_orderkey
and c_custkey = o_custkey
and s_nationkey = n1.n_nationkey
and c_nationkey = n2.n_nationkey
and (
(n1.n_name = 'FRANCE' and n2.n_name = 'GERMANY')
or (n1.n_name = 'GERMANY' and n2.n_name = 'FRANCE')
)
and l_shipdate between to_date( '1995-01-01', 'YYYY-MM-DD') and
to_date( '1996-12-31', 'YYYY-MM-DD')
) shipping
group by
supp_nation,
cust_nation,
l_year
order by
supp_nation,
cust_nation,
l_year

SUPP_NATION          CUST_NATION          L_YEAR
REVENUE
FRANCE               GERMANY              1995.00
54639732.73
FRANCE               GERMANY              1996.00
54633083.31
GERMANY              FRANCE               1995.00
52531746.67
GERMANY              FRANCE               1996.00
52520549.02


4 rows processed.
Query Processed in 3.59 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:22 2004


Stream Started at 1074908358.88
Stream Ended at 1074908362.47
Stream Processed in 3.59 seconds


SQL statements processed: 1
```

8.log

```
Begin Execution at Fri Jan 23 19:39:22 2004



-- using default substitutions


select
o_year,
sum(case when nation='BRAZIL' then volume else 0 end )/
sum(volume)
as mkt_share
from
(
select
to_number (to_char (o_orderdate, 'yyyy')) as o_year,
l_extendedprice * (1 - l_discount) as volume,
n2.n_name as nation
from
part,
supplier,
lineitem,
orders,
customer,
nation n1,
nation n2,
region
where
p_partkey = l_partkey
and s_suppkey = l_suppkey
and l_orderkey = o_orderkey
and o_custkey = c_custkey
and c_nationkey = n1.n_nationkey
and n1.n_regionkey = r_regionkey
and r_name = 'AMERICA'
and s_nationkey = n2.n_nationkey
and o_orderdate between to_date ('1995-01-01', 'YYYY-MM-DD') and
to_date ('1996-12-31', 'YYYY-MM-DD')
and p_type = 'ECONOMY ANODIZED STEEL'
) all_nations
group by
o_year
order by
o_year

O_YEAR               MKT_SHARE
1995.00              0.03
1996.00              0.04


2 rows processed.
Query Processed in 4.50 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:27 2004


Stream Started at 1074908362.55
Stream Ended at 1074908367.04
Stream Processed in 4.50 seconds


SQL statements processed: 1
```

9.log

```
_____________________________________
Begin Execution at Fri Jan 23 19:39:27 2004


-- using default substitutions


select
nation,
o_year,
sum(amount) as sum_profit
from
(
select
n_name as nation,
to_number (to_char (o_orderdate, 'yyyy')) as o_year,
l_extendedprice * (1 - l_discount) - ps_supplycost * l_quantity as
amount
from
part,
supplier,
                                        lineitem,
partsupp,
orders,
                                        nation
where
s_suppkey = l_suppkey
and ps_suppkey = l_suppkey
and ps_partkey = l_partkey
and p_partkey = l_partkey
and o_orderkey = l_orderkey
and s_nationkey = n_nationkey
and p_name like '%green%'
) profit
group by
nation,
o_year
order by
nation,
o_year desc

NATION                O_YEAR              SUM_PROFIT
ALGERIA               1998.00             31342867.23
ALGERIA               1997.00             57138193.02
ALGERIA               1996.00             56140140.13
ALGERIA               1995.00             53051469.65
ALGERIA               1994.00             53867582.13
ALGERIA               1993.00             54942718.13
ALGERIA               1992.00             54628034.71



<.................lines deleted .................>



VIETNAM               1998.00             30442736.06
VIETNAM               1997.00             50309179.79
VIETNAM               1996.00             50488161.41
VIETNAM               1995.00             49658284.61
VIETNAM               1994.00             50596057.26
VIETNAM               1993.00             50953919.15
VIETNAM               1992.00             49613838.32


175 rows processed.
Query Processed in 8.61 seconds.

Ended Executing this Stream at Fri Jan 23 19:39:35 2004


Stream Started at 1074908367.12
Stream Ended at 1074908375.73
Stream Processed in 8.61 seconds


SQL statements processed: 1


_____________________________________
            10.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:35 2004


-- using default substitutions


select  *  from  (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
customer,
orders,
lineitem,
nation
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate >= to_date ('1993-10-01',  'YYYY-MM-DD')
and o_orderdate < add_months(  to_date( '1993-10-01',  'YYYY-MM-
DD'),  3)
and l_returnflag = 'R'
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20

C_CUSTKEY             C_NAME                    REVENUE
C_ACCTBAL             N_NAME
C_ADDRESS                          C_PHONE
C_COMMENT
57040.00              Customer#000057040        734235.25
632.87                JAPAN
Eioyzjf4pp                         22-895-641-3466
requests sleep blithely about the furiously i
143347.00             Customer#000143347        721002.69
2557.47               EGYPT
1aReFYv,Kw4                        14-742-935-3718
```

```
fluffily bold excuses haggle finally after the u


<.....lines deleted .....>


551.79          ARGENTINA
NFztyTOR10UOJ                   11-208-192-3205
unusual requests detect. slyly dogged theodolites use slyly. deposit
23431.00        Customer#000023431      554269.54
3381.86         ROMANIA
HgiV0phqhaIa9aydNoIlb           29-915-458-2654
instructions nag quickly. furiously bold accounts cajol


20 rows processed.
Query Processed in 6.52 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:42 2004


Stream Started at 1074908375.82
Stream Ended at 1074908382.34
Stream Processed in 6.52 seconds


SQL statements processed: 1


_____________________________________
        11.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:42 2004


-- using default substitutions


select
ps_partkey,
sum(ps_supplycost * ps_availqty) as value
from
partsupp,
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'GERMANY'
group by
ps_partkey having
sum(ps_supplycost * ps_availqty) > (
select
sum(ps_supplycost * ps_availqty) * 0.0001000000
from
partsupp,
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'GERMANY'
)
order by
value desc


PS_PARTKEY          VALUE
129760.00           17538456.86
166726.00           16503353.92
191287.00           16474801.97
161758.00           16101755.54
34452.00            15983844.72
139035.00           15907078.34
72922.00            7930400.64


<.....lines deleted .....>


128820.00           7892882.72
25891.00            7890511.20
122819.00           7888881.02
154731.00           7888301.33
101674.00           7879324.60
51968.00            7879102.21
72073.00            7877736.11
5182.00             7874521.73


1048 rows processed.
Query Processed in 2.97 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:45 2004


Stream Started at 1074908382.42
Stream Ended at 1074908385.38
Stream Processed in 2.97 seconds


sQL statements processed: 1


_____________________________________
        12.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:45 2004


-- using default substitutions


select
        l_shipmode,
        sum(case
                when o_orderpriority = '1-URGENT'
                        or o_orderpriority = '2-HIGH'
                        then 1
                else 0
        end) as high_line_count,
        sum(case
                when o_orderpriority <> '1-URGENT'
                        and o_orderpriority <> '2-HIGH'
                        then 1
                else 0
        end) as low_line_count
from
        orders,
        lineitem
where
        o_orderkey = l_orderkey
```

```
        and l_shipmode in ('MAIL', 'SHIP')
        and l_commitdate < l_receiptdate
        and l_shipdate < l_commitdate
and l_receiptdate >= to_date( '1994-01-01',  'YYYY-MM-DD')
and l_receiptdate < add_months(to_date ('1994-01-01',  'YYYY-MM-
DD'),  12)
group by
        l_shipmode
order by
        l_shipmode

L_SHIPMODE HIGH_LINE_COUNT      LOW_LINE_COUNT
MAIL    6202.00         9324.00
SHIP    6200.00         9262.00


2 rows processed.
Query Processed in 3.52 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:48 2004


Stream Started at 1074908385.46
Stream Ended at 1074908388.98
Stream Processed in 3.52 seconds


SQL statements processed: 1


_____________________________________
        13.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:49 2004


-- using default substitutions


select
c_count,
count(*) as custdist
from
(
select
c_custkey,
count(o_orderkey)  as  c_count
from
customer, orders where
c_custkey = o_custkey(+)
and o_comment(+) not like '%special%requests%'
group by
c_custkey
) c_orders
group by
c_count
order by
custdist desc,
c_count desc

C_COUNT         CUSTDIST
0.00            50004.00
9.00            6641.00
10.00           6566.00
11.00           6058.00
8.00            5949.00
12.00           5553.00
13.00           4989.00
19.00           4748.00


<.............lines deleted .....................>


33.00           71.00
34.00           48.00
35.00           33.00
1.00            23.00
36.00           17.00
37.00           7.00
40.00           4.00
38.00           4.00
39.00           2.00
41.00           1.00


42 rows processed.
Query Processed in 5.76 seconds.


Ended Executing this Stream at Fri Jan 23 19:39:54 2004


Stream Started at 1074908389.06
Stream Ended at 1074908394.82
Stream Processed in 5.76 seconds


SQL statements processed: 1


_____________________________________
        14.log
_____________________________________
Begin Execution at Fri Jan 23 19:39:54 2004


-- using default substitutions


select
        100.00 * sum(case
                when p_type like 'PROMO%'
                        then l_extendedprice * (1 - l_discount)
                else 0
        end) / sum(l_extendedprice * (1 - l_discount)) as
promo_revenue
from
        lineitem,
        part
where
        l_partkey = p_partkey
        and l_shipdate >= date '1995-09-01'
        and l_shipdate < date '1995-09-01' + interval '1' month

PROMO_REVENUE
16.38


1 row processed.
Query Processed in 0.87 seconds.
```

Ended Executing this Stream at Fri Jan 23 19:39:55 2004

Stream Started at 1074908394.90
Stream Ended at 1074908395.76
Stream Processed in 0.87 seconds

SQL statements processed: 1

_____
15.log
_____
Begin Execution at Fri Jan 23 19:39:55 2004

-- using default substitutions

```
create view revenue0 (supplier_no, total_revenue) as
select
l_suppkey,
sum(l_extendedprice * (1 - l_discount))
from
lineitem
where
l_shipdate >= to_date( '1996-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1996-01-01', 'YYYY-MM-DD'),
3)
group by
l_suppkey
```

Query Processed in 0.02 seconds.

```
select
s_suppkey,
s_name,
s_address,
s_phone,
total_revenue
from
supplier,
revenue0
where
s_suppkey = supplier_no
and total_revenue = (
select
max(total_revenue)
from
revenue0
)
order by
s_suppkey
```

S_SUPPKEY S_NAME
S_ADDRESS S_PHONE TOTAL_REVENUE
8449.00 Supplier#000008449
Wp34zim9qYFbVctdW 20-469-856-8873 1772627.21

1 row processed.
Query Processed in 1.24 seconds.

```
drop view revenue0
```

Query Processed in 0.54 seconds.

Ended Executing this Stream at Fri Jan 23 19:39:57 2004

Stream Started at 1074908395.84
Stream Ended at 1074908397.64
Stream Processed in 1.80 seconds

SQL statements processed: 3

_____
16.log
_____
Begin Execution at Fri Jan 23 19:39:57 2004

-- using default substitutions

```
select
p_brand,
p_type,
p_size,
count(distinct ps_suppkey) as supplier_cnt
from
partsupp,
part
where
p_partkey = ps_partkey
and p_brand <> 'Brand#45'
and p_type not like 'MEDIUM POLISHED%'
and p_size in (49, 14, 23, 45, 19, 3, 36, 9)
and ps_suppkey not in (
select
s_suppkey
from
supplier
where
s_comment like '%Customer%Complaints%'
)
group by
p_brand,
p_type,
p_size
order by
supplier_cnt desc,
p_brand,
p_type,
p_size
```

P_BRAND P_TYPE P_SIZE SUPPLIER_CNT
Brand#41 MEDIUM BRUSHED TIN 3.00 28.00
Brand#54 STANDARD BRUSHED COPPER 14.00 27.00
Brand#11 STANDARD BRUSHED TIN 23.00 24.00
Brand#11 STANDARD BURNISHED BRASS 36.00 24.00
Brand#15 MEDIUM ANODIZED NICKEL 3.00 24.00
Brand#15 SMALL ANODIZED BRASS 45.00 24.00
Brand#15 SMALL BURNISHED NICKEL 19.00 24.00

<.....lines deleted .....>

```
Brand#35   MEDIUM ANODIZED TIN       19.00              3.00
Brand#51   SMALL PLATED BRASS       23.00              3.00
Brand#52   MEDIUM BRUSHED BRASS      45.00              3.00
Brand#53   MEDIUM BRUSHED TIN       45.00              3.00
Brand#54   ECONOMY POLISHED BRASS    9.00              3.00
Brand#55   PROMO PLATED BRASS       19.00              3.00
Brand#55   STANDARD PLATED TIN      49.00              3.00


18314 rows processed.
Query Processed in 2.46 seconds.


Ended Executing this Stream at Fri Jan 23 19:40:00 2004


Stream Started at 1074908397.72
Stream Ended at 1074908400.17
Stream Processed in 2.46 seconds


SQL statements processed: 1


_____________________________________
          17.log
_____________________________________
Begin Execution at Fri Jan 23 19:40:00 2004



-- using default substitutions


select
sum(l_extendedprice) / 7.0 as avg_yearly
from
lineitem,
part
where
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container = 'MED BOX'
and l_quantity < (
select
0.2 * avg(l_quantity)
from
lineitem
where
l_partkey = p_partkey
)

AVG_YEARLY
348406.05


1 row processed.
Query Processed in 4.42 seconds.


Ended Executing this Stream at Fri Jan 23 19:40:04 2004


Stream Started at 1074908400.25
Stream Ended at 1074908404.67
Stream Processed in 4.42 seconds


SQL statements processed: 1


_____________________________________
          18.log
_____________________________________
Begin Execution at Fri Jan 23 19:40:04 2004



-- using default substitutions


select * from (
select
c_name,
c_custkey,
o_orderkey,
o_orderdate,
o_totalprice,
sum(l_quantity)
from
customer,
orders,
lineitem
where
o_orderkey in (
select
l_orderkey
from
lineitem
group by
l_orderkey having
sum(l_quantity) > 300
)
and c_custkey = o_custkey
and o_orderkey = l_orderkey
group by
c_name,
c_custkey,
o_orderkey,
o_orderdate,
o_totalprice
order by
o_totalprice desc,
o_orderdate
)
where rownum <= 100

C_NAME                C_CUSTKEY           O_ORDERKEY
O_ORDERDATE
O_TOTALPRICE          SUM(L_QUANTITY)
Customer#000128120       128120.00           4722021.00           1994-
04-07
544089.09             323.00
Customer#000144617       144617.00           3043270.00           1997-
02-12
530604.44             317.00
Customer#000013940       13940.00            2232932.00           1997-
04-13
522720.61             304.00
Customer#000066790       66790.00            2199712.00           1996-
09-30
515531.82             327.00
Customer#000046435       46435.00            4745607.00           1997-
07-03
508047.99             309.00
Customer#000015272       15272.00            3883783.00           1993-
07-28
500241.33             302.00
```

```
Customer#000146608      146608.00       3342468.00      1994-
06-12


<.....lines deleted .....>


Customer#000017746      17746.00        6882.00         1997-
04-09
408446.93       303.00
Customer#000013072      13072.00        1481925.00      1998-
03-15
399195.47       301.00
Customer#000082441      82441.00        857959.00       1994-
02-07
382579.74       305.00
Customer#000088703      88703.00        2995076.00      1994-
01-30
363812.12       302.00


57 rows processed.
Query Processed in 8.65 seconds.


Ended Executing this Stream at Fri Jan 23 19:40:13 2004


Stream Started at 1074908404.75
Stream Ended at 1074908413.40
Stream Processed in 8.65 seconds


SQL statements processed: 1


_____________________________________
          19.log
_____________________________________
Begin Execution at Fri Jan 23 19:40:13 2004


-- using default substitutions


select
sum(l_extendedprice* (1 - l_discount)) as revenue
from
lineitem,
part
where
(
p_partkey = l_partkey
and p_brand = 'Brand#12'
and p_container in ('SM CASE', 'SM BOX', 'SM PACK', 'SM PKG')
and l_quantity >= 1 and l_quantity <= 1 + 10
and p_size between 1 and 5
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container in ('MED BAG', 'MED BOX', 'MED PKG', 'MED
PACK')
and l_quantity >= 10 and l_quantity <= 10 + 10
and p_size between 1 and 10
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)
or
(
p_partkey = l_partkey
and p_brand = 'Brand#34'
and p_container in ('LG CASE', 'LG BOX', 'LG PACK', 'LG PKG')
and l_quantity >= 20 and l_quantity <= 20 + 10
and p_size between 1 and 15
and l_shipmode in ('AIR', 'AIR REG')
and l_shipinstruct = 'DELIVER IN PERSON'
)

REVENUE
3083843.06


1 row processed.
Query Processed in 3.96 seconds.


Ended Executing this Stream at Fri Jan 23 19:40:17 2004


Stream Started at 1074908413.48
Stream Ended at 1074908417.44
Stream Processed in 3.96 seconds


SQL statements processed: 1


_____________________________________
          20.log
_____________________________________
Begin Execution at Fri Jan 23 19:40:17 2004


-- using default substitutions


select
s_name,
s_address
from
supplier,
nation
where
s_suppkey in (
select
ps_suppkey
from
partsupp
where
ps_partkey in (
select
p_partkey
from
part
where
p_name like 'forest%'
)
and ps_availqty > (
select
0.5 * sum(l_quantity)
from
lineitem
```

```
where
l_partkey = ps_partkey
and l_suppkey = ps_suppkey
and l_shipdate >= to_date ('1994-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1994-01-01', 'YYYY-MM-DD'),
12)
)
)
and s_nationkey = n_nationkey
and n_name = 'CANADA'
order by
s_name

S_NAME                S_ADDRESS
Supplier#000000020        iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197        YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226        83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285        Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378        FfbhyCxWvcPrO8ltp9
Supplier#000000402        i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530        0qwCMwobKY OcmLyfRXlagA8ukENJv,



<.....lines deleted .....>



Supplier#000009709        rRnCbHYgDgl9PZYnyWKVYSUW0vKg
Supplier#000009753        wLhVEcRmd7PkJF4FBnGK7Z
Supplier#000009796        z,y4Idmr15DOvPUqYG
Supplier#000009799         4wNjXGa4OKWl
Supplier#000009811        E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812        APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862        rJzweWeN58
Supplier#000009868        ROjGgx5gvtkmnUUoeyy7v
Supplier#000009869
ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
Supplier#000009899        7XdpAHrzr1t,UQFZE
Supplier#000009974        7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT


204 rows processed.
Query Processed in 4.08 seconds.


Ended Executing this Stream at Fri Jan 23 19:40:21 2004


Stream Started at 1074908417.51
Stream Ended at 1074908421.59
Stream Processed in 4.08 seconds


SQL statements processed: 1


_____________________________________
          21.log
_____________________________________
Begin Execution at Fri Jan 23 19:40:21 2004


-- using default substitutions

select * from (
select
s_name,
count(*)  numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.l_receiptdate > l1.l_commitdate
and exists (
select
*
from
lineitem l2
where
l2.l_orderkey = l1.l_orderkey
and l2.l_suppkey <> l1.l_suppkey
)
and not exists (
select
*
from
lineitem l3
where
l3.l_orderkey = l1.l_orderkey
and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate
)
and s_nationkey = n_nationkey
and n_name = 'SAUDI ARABIA'
group by
s_name
order by
numwait desc,
s_name)
where rownum <= 100

S_NAME                NUMWAIT
Supplier#000002829        20.00
Supplier#000005808        18.00
Supplier#000000262        17.00
Supplier#000000496        17.00
Supplier#000002160        17.00
Supplier#000002301        17.00
Supplier#000002540        17.00
Supplier#000003063        17.00



<.....lines deleted .....>



Supplier#000000821        12.00
Supplier#000001337        12.00
Supplier#000001916        12.00
Supplier#000001925        12.00
Supplier#000002039        12.00
Supplier#000002357        12.00
Supplier#000002483        12.00
```

100 rows processed.
Query Processed in 15.91 seconds.

Ended Executing this Stream at Fri Jan 23 19:40:37 2004

Stream Started at 1074908421.67
Stream Ended at 1074908437.58
Stream Processed in 15.91 seconds

SQL statements processed: 1

_____
22.log
_____
Begin Execution at Fri Jan 23 19:40:37 2004

-- using default substitutions

```
select
cntrycode,
count(*) as numcust,
sum(c_acctbal) as totacctbal
from
(
select
substr(c_phone, 1, 2) as cntrycode,
c_acctbal
from
customer
where
substr(c_phone,1, 2) in
('13', '31', '23', '29', '30', '18', '17')
and c_acctbal > (
select
avg(c_acctbal)
from
customer
where
c_acctbal > 0.00
and substr(c_phone, 1, 2) in
('13', '31', '23', '29', '30', '18', '17')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode
```

CNTRYCODE	NUMCUST	TOTACCTBAL
13	888.00	6737713.99
17	861.00	6460573.72
18	964.00	7236687.40
23	892.00	6701457.95
29	948.00	7158866.63
30	909.00	6808436.13
31	922.00	6806670.18

7 rows processed.
Query Processed in 2.67 seconds.

Ended Executing this Stream at Fri Jan 23 19:40:40 2004

Stream Started at 1074908437.66
Stream Ended at 1074908440.32
Stream Processed in 2.67 seconds

SQL statements processed: 1

Appendix E: Seed and input parameters

Seed

117143112

qp1.0

```
14 1994-08-01
2 45 TIN AFRICA
9 moccasin
20 dodger 1996-01-01 SAUDI ARABIA
6 1996-01-01 0.06 25
17 Brand#33 JUMBO PACK
18 314
8 UNITED KINGDOM EUROPE LARGE PLATED
COPPER
21 ROMANIA
13 unusual requests
3 HOUSEHOLD 1995-03-06
22 17 22 10 12 11 29
23
16 Brand#33 LARGE ANODIZED 9 14 39
42 26 34 47 25
4 1993-11-01
11 FRANCE 0.0000000333
15 1993-06-01
1 111
10 1994-09-01
19 Brand#42 Brand#45 Brand#35 9 15 22
5 ASIA 1996-01-01
7 KENYA UNITED KINGDOM
12 MAIL RAIL 1993-01-01
```

qp1.1

```
21 IRAQ
3 BUILDING 1995-03-22
18 315
5 EUROPE 1996-01-01
11 ROMANIA 0.0000000333
7 ETHIOPIA MOROCCO
6 1996-01-01 0.03 25
20 peach 1995-01-01 IRAN
17 Brand#35 JUMBO DRUM
12 TRUCK MAIL 1994-01-01
16 Brand#13 PROMO PLATED 40 26 14
25 48 34 33 28
15 1996-01-01
13 unusual requests
10 1993-06-01
2 33 COPPER EUROPE
8 MOROCCO AFRICA LARGE ANODIZED
COPPER
14 1994-11-01
19 Brand#44 Brand#33 Brand#34 4 16 29
9 maroon
22 11 23 28 30 10 29
14
1 119
4 1996-06-01
```

qp1.2

```
6 1997-01-01 0.09 24
17 Brand#32 WRAP BOX
14 1995-02-01
16 Brand#53 SMALL POLISHED 29 5 31
43 10 16 41 32
19 Brand#51 Brand#11 Brand#23 9 17 26
10 1994-03-01
9 lawn
2 20 STEEL AMERICA
15 1993-09-01
8 GERMANY EUROPE MEDIUM POLISHED
COPPER
5 MIDDLE EAST 1997-01-01
22 10 19 12 31 25 23
22
12 AIR MAIL 1995-01-01
7 RUSSIA GERMANY
13 unusual requests
18 313
1 66
4 1994-03-01
20 blush 1993-01-01 ALGERIA
3 HOUSEHOLD 1995-03-08
11 GERMANY 0.0000000333
21 CANADA
```

qp1.3

```
8 UNITED STATES AMERICA MEDIUM
BURNISHED COPPER
5 AFRICA 1997-01-01
4 1996-10-01
6 1997-01-01 0.06 25
17 Brand#34 WRAP PACK
7 KENYA UNITED STATES
1 74
18 314
22 25 17 19 12 14 16
22
14 1995-05-01
9 hot
10 1995-01-01
15 1996-04-01
11 SAUDI ARABIA 0.0000000333
20 magenta 1997-01-01 MOROCCO
2 8 BRASS EUROPE
21 SAUDI ARABIA
19 Brand#53 Brand#54 Brand#22 4 18 22
13 unusual requests
```

```
16      Brand#33 ECONOMY ANODIZED        46      26
        42      31      33      19      12      39
12      REG AIR FOB     1995-01-01
3       AUTOMOBILE      1995-03-24
```

qp1.4

```
5       AMERICA         1997-01-01
21      JAPAN
14      1995-08-01
19      Brand#55 Brand#31 Brand#22 10      19      29
15      1994-01-01
17      Brand#31 WRAP DRUM
12      SHIP    FOB     1995-01-01
6       1997-01-01      0.04    25
4       1994-07-01
9       gainsboro
8       MOZAMBIQUE      AFRICA  MEDIUM ANODIZED
TIN
16      Brand#13 STANDARD BURNISHED      50      3
        27      32      12      2       22      36
11      INDIA   0.0000000333
2       46      NICKEL  AMERICA
10      1993-10-01
18      312
1       82
13      express accounts
7       FRANCE  MOZAMBIQUE
22      23      16      11      28      31      10
        13
3       HOUSEHOLD       1995-03-10
20      thistle 1995-01-01      EGYPT
```

qp1.5

```
21      EGYPT
15      1996-07-01
4       1997-01-01
6       1997-01-01      0.09    24
7       UNITED KINGDOM INDIA
16      Brand#53 MEDIUM POLISHED         3       42
        20      22      15      41      49      8
19      Brand#12 Brand#14 Brand#11 5       20      25
18      313
14      1995-12-01
22      29      20      13      24      33      22
        18
11      VIETNAM         0.0000000333
13      express accounts
3       AUTOMOBILE      1995-03-26
1       90
2       34      COPPER  MIDDLE EAST
5       ASIA    1997-01-01
8       INDIA   ASIA    SMALL POLISHED TIN
20      goldenrod 1993-01-01    ROMANIA
12      FOB     TRUCK   1996-01-01
17      Brand#33 SM BAG
10      1994-07-01
9       dodger
```

qp1.6

```
10      1993-04-01
3       FURNITURE       1995-03-12
15      1994-04-01
13      express accounts
6       1993-01-01      0.07    24
8       ALGERIA AFRICA  SMALL BURNISHED TIN
9       cornsilk
7       MOROCCO         ALGERIA
4       1994-10-01
11      INDONESIA       0.0000000333
22      17      22      29      18      15      12
        28
18      315
12      MAIL    FOB     1996-01-01
1       98
5       MIDDLE EAST     1993-01-01
16      Brand#33 ECONOMY BRUSHED         18      39
        8       16      12      23      15      45
2       21      STEEL   AMERICA
14      1996-03-01
19      Brand#14 Brand#52 Brand#15 10      10      21
20      rose    1997-01-01      INDONESIA
17      Brand#35 SM PACK
21      RUSSIA
```

qp1.7

```
18      312
8       PERU    AMERICA         STANDARD BRUSHED
TIN
20      cornflower 1995-01-01   UNITED STATES
21      KENYA
2       9       BRASS   MIDDLE EAST
4       1997-05-01
22      32      24      13      14      21      31
        20
17      Brand#32 SM DRUM
1       106
11      RUSSIA  0.0000000333
9       burnished
19      Brand#11 Brand#35 Brand#14 5       11      28
3       MACHINERY       1995-03-28
13      express accounts
5       AFRICA  1993-01-01
7       GERMANY         PERU
10      1994-02-01
16      Brand#13 SMALL BURNISHED         17      27
        18      49      30      23      25      8
6       1993-01-01      0.04    25
14      1996-06-01
15      1996-11-01
12      RAIL    SHIP    1996-01-01
```

qp1.8

```
19      Brand#24 Brand#23 Brand#53 1       12      25
1       114
15      1994-08-01
17      Brand#33 LG BAG
5       AMERICA         1993-01-01
8       INDONESIA       ASIA    STANDARD PLATED
TIN
9       black
12      AIR     SHIP    1996-01-01
14      1996-09-01
7       UNITED STATES   INDONESIA
4       1995-02-01
3       FURNITURE       1995-03-14
```

20 navajo 1994-01-01 JORDAN
16 Brand#53 LARGE PLATED 43 7 9
28 47 45 31 12
6 1993-01-01 0.09 24
22 13 27 17 23 15 14
12
10 1994-11-01

13 express accounts
2 47 NICKEL ASIA
21 FRANCE
18 314
11 IRAN 0.0000000333

Appendix F: Benchmark Scripts

_____
2ocssd
_____

```
#!/bin/ksh

. $FRAME_PATH/env
if [[ $1 == "start" ]]; then
  for node in $ALL_NODES; do
   rsh $node ocssd_start
  done
elif [[ $1 == "strace" ]]; then
  for node in $ALL_NODES; do
   rsh $node ocssd_strace $2
  done
elif [[ $1 == "stop" ]]; then
    for node in $ALL_NODES; do
      echo stopping cluster manager on node $node
      rsh $node ocssd_stop
    done
fi
```

_____
2shut
_____

```
#!/bin/ksh
. $FRAME_PATH/env

if [ "$1" = "abort" ]; then
for i in $SECONDARY_NODES
do
rsh $i -n $KIT_DIR/rshuta
done
sqlplus << !
connect / as sysdba
shutdown abort
exit
!
else
for i in $SECONDARY_NODES
do
rsh $i -n $KIT_DIR/rshut
done
sqlplus << !
connect / as sysdba
shutdown immediate
exit
!
fi
```

_____
2start
_____

```
#!/bin/ksh

. $FRAME_PATH/env
cud=`pwd`
cd $ORACLE_HOME/bin
cpall oracle

cd /home/oracle/tpch/dbs
cpall init_run.ora
cd $cud

if [[ "$1" = "" ]]; then
  nodes=$ALL_NODES
else
  nodes=$1
fi
for i in $nodes
do
echo Bringing up node $i
rsh $i -n $KIT_DIR/rstart &
done
wait
echo Nodes are up!
```

_____
gen_seed.sh
_____

```
#!/bin/ksh

SEED_FILE=$1

#Generate the seed
echo "Setting the random number seed"
PSEED=`date +%m:%d:%H:%M:%S | sed -e 's/://g'`
echo "Using ${PSEED} as seed0"
echo ${PSEED} >  $SEED_FILE
echo "Done setting the random number seed"
```

_____
ocssd_start
_____

```
#!/bin/ksh

. $FRAME_PATH/env

#need to get corefiles
cd $ORACLE_HOME/css/init
ulimit -c unlimited

node=`uname -a | awk '{print $2}'`
cup=0
cup=`ps -ef | grep " ocssd.bin" | grep -v grep | awk '{print $2}'`
if [[ $cup == "" ]]; then
   echo starting cluster manager on $node
   nohup ocssd.bin &
else
   echo cluster manager already up on $node
fi
```

_____
ocssd_stop
_____

```
#!/bin/ksh

. $FRAME_PATH/env
node=`uname -a | awk '{print $2}'`
cup=0
```

```
cup=`ps -ef | grep " ocssd.bin" | grep -v grep | awk '{print $2}' | wc -l`
if [[ $cup -ge 1 ]]; then
   echo stopping cluster manager on $node
   ps -ef | grep " ocssd.bin" | grep -v grep | awk '{print $2}' | xargs kill -9
else
   echo cluster manager already down on $node
fi
```

qexecpl.c

```
#ifdef RCSID
static char *RCSid =
  "$Header: qexecpl.c 17-oct-2001.09:29:47 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.  */

/*

  NAME
    qexecpl.c - <one-line expansion of the name>

  DESCRIPTION
    SQL Execution Engine, Oracle v8,  OCI version

  PRIVATE FUNCTION(S)
    <list of static functions defined in .c file - with one-line descriptions>

  MODIFIED   (MM/DD/YY)
    mpoess    10/17/01 - add serialization level in SQLinit
    mpoess    02/22/01 - add linux changes
    mpoess    08/05/99 - make compile
    mpoess    11/13/98 - fix pddl statement
    pswong    02/19/97 - migrating to version 8
    pswong    04/02/96 - more polishing
    pswong    03/25/96 - polish up
    pswong    03/06/96 - created

*/

#include <stdio.h>
#include <string.h>
#include <setjmp.h>
#include <sys/param.h>
#include <errno.h>
#include <math.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>
#include <stdlib.h>

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

int define_output_variables();
void process_select_list();
void usage();
void SQLinit();
void SQLexec();
void SQLexit();
void *memalloc();
void print_header();
void print_rows();
int OFEN();
void remove_newline();

char logname[UNAME_LEN];  /* username/passwd combo */
char *passwd;

double tr_start = 0.0;    /* query start time                */
double tr_end = 0.0;      /* query end time                  */

double s_tr_start = 0.0;  /* statement start time            */
double s_tr_end = 0.0;    /* statement end time              */

/* For our purpose of timing, we will treat comments as delimiters */
/* for queries.   Thus, we will collect query timings whenever we  */
/* encounter a comment (of course not for the first comment in a   */
/* file).                                                          */

int end_flag = 0;         /* flag to indicate that we have reached */
                          /* the end of a query                    */

int stmt_cnt = 0;         /* Number of statements processed.     */
int qry_cnt = 0;          /* Number of query processed.          */

double product = 1.0;     /* cumulative product of query times   */
int rows_ret = 0;         /* the number of rows fetched          */
int num_sel_list = 0;     /* the number of select list item      */

long num_to_fetch = -1;   /* Number of rows to fetch. -1 means fetch all
*/

sltype slist[MAX_SEL_LIST];  /* Array for describing Select List
*/
dltype *dlist[MAX_SEL_LIST]; /* Array of ptrs for Defining Select
List   */

char stmt[SQL_LEN];       /* The SQL statement or comment line.  */
char qn[3];               /* Number of the query being executed  */
char qnp[3];              /* Number of the previous query executed  */
char cmnt[5000];          /* Buffer to save the comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template               */
FILE *logfile;   /* log and report files                */
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template               */
FILE *logfile = stdout;   /* log and report files                */
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN           */
int deflen = 0;           /* Size of data type for ODEFIN        */
int deftype = 1;          /* Oracle type number for ODEFIN       */

int pfmem = PFMEMSIZE;    /* Memory to prefetch rows             */

time_t tim;               /* To get wall clock time              */

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
```

```
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curq = NULL;
OCIStmt *cur_dml = NULL;
OCIStmt *cur_ddl = NULL;
OCIParam *tpcpar = NULL;

sword status = OCI_SUCCESS;  /* OCI return value */

/* usage: prints the usage of the program */

void usage() {

  fprintf(stderr,"\nUsage: qexec username/password [q<path name for
query template file>]\n");
  fprintf(stderr,"               [l<path name for log>] [r<path name for
reports>]\n\n");
  fprintf(stderr,"Options:\n");
  fprintf(stderr,"q<path for query>         : full path name for the query
template file.\n");
  fprintf(stderr,"                          (default is stdin)\n");
  fprintf(stderr,"l<path name for log>      : full path name for log
files\n");
  fprintf(stderr,"                          (default is stdout)\n");
  fprintf(stderr,"r<path name for reports>  : full path name for
reports\n");
  fprintf(stderr,"                          (default is stdout)\n");
  exit(-1);
}



/* type: 0 if environment handle is passed, 1 if error handle is passwd */

void sql_error(errhp,status,type)
     OCIError *errhp;
     sword status;
     sword type;
{
  char msg[2048];
  ub4 errcode;
  ub4 msglen;
  int i,j;

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement returned with info.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_ERROR:
    fprintf(stderr, "Error: OCI call error.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }

  /* Rollback just in case */

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

  fprintf(stderr, "Exiting Oracle...\n");
  fflush(stderr);

  SQLexit();

  exit(1);
}

#ifdef LINUX
int main(argc,argv)
#else
void main(argc,argv)
#endif
     int argc;
     char *argv[];
{

  int i,pos,pos2;
  int retcode;      /* Return code for get_statement */
#ifdef LINUX
  logfile=fopen("/dev/stdout","w");
  qtemp=fopen("/dev/stdin","rw");
  rep=fopen("/dev/stdout","w");
#endif
  /* Initialize some variables */

  if ((argc > 5) || (argc < 2)) {
    usage();
  }

  /* argv[1] -- User and Password for Database */

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'q':
      if ((qtemp = fopen(++(argv[0]),"r")) == NULL) {
                fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
                fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
                exit(-1);
      }
      break;
    case 'r':
      if ((rep = fopen(++(argv[0]),"a")) == NULL) {
                fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
                fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
                exit(-1);
      }
      break;
    case 'l':
      if ((logfile = fopen(++(argv[0]),"a")) == NULL) {
                fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
```

```
            fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
            exit(-1);
    }
    break;
  default:
    fprintf(stderr,"Invalid Option: %c\n", argv[0][0]);
    usage();
    break;
  }
}

/* Do some initialization and establish connection with the database */

SQLinit();

/* May want to add some triggering mechanism here */

time(&tim);
fprintf(logfile, "Begin Execution at %s\n\n", ctime(&tim));
fprintf(rep, "Begin Executing this Stream at %s\n\n", ctime(&tim));
/* Get the next statement and start processing it */

while ((retcode = get_statement()) > 0) {

  switch (retcode) {

    /* If this is a comment, skips it */
  case COMMENT:
    /*if (end_flag) {
            end_flag = 0;  */    /* reset query end flag */
            /* save the comment so that we can print it out later on  */
            /* strcpy(cmnt, stmt);
            break;
    } */
    if (stmt[3]== '@') {
      pos=4;
      strcpy(qnp,qn);
      while (stmt[pos] != ')') {
        pos++;
      }
      pos2=0;
      pos++;
      while (stmt[pos] != '.') {
        /*printf ("qn %d %c \n",pos2,stmt[pos]);*/
        qn[pos2]=stmt[pos];
        pos2++;
        pos++;
      }
      qn[pos2] = 0;
      /* printf("found a new query: %s\n",qn); */
    }
    /* save the comment so that we can print it out later on  */
    strcat(cmnt, stmt);
    break;

    /* if this is a set_row_fetch command */
  case SET_FETCHROW:
    fprintf(logfile,"Setting the number of rows to fetch to: %ld\n\n",
                num_to_fetch);
    break;

    /* if this is a SQL statement */
  case SQL_STMT:

    /* Executes the query */
    SQLexec();

    stmt_cnt++;
    qry_cnt++;
    fflush(rep);
    fflush(logfile);
    /*
    fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
    fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);

    fprintf(logfile,"Statement Processed in %.2f seconds.\n",
                (s_tr_end - s_tr_start));
    fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
                qn,(s_tr_end - s_tr_start)s_tr_start,s_tr_end);
    fflush(rep);
    fflush(logfile);*/
    break;

    /* Should never reach here */
  default:
    fprintf(stderr, "Invalid statement type!!\n");
    SQLexit();
    break;
  }
}

  /* Get Timing for the last query */

  tr_end = gettime();

  fprintf(logfile,"Query Processed in %.2f seconds.\n\n",(tr_end -
s_tr_start));

  /* print comments for this query that we have saved */

  /* fprintf(logfile, "%s\n", cmnt); */

  /* fprintf(rep, "Query %s : Execution time %.2f\n", qn,(tr_end -
s_tr_start));*/
  fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
                qn,(tr_end - s_tr_start),s_tr_start,tr_end);

  time(&tim);
  fprintf(logfile,"\nEnded Executing this Stream at %s\n", ctime(&tim));
  fprintf(logfile,"\nStream Started at %.2f\n", tr_start);
  fprintf(logfile,"Stream Ended at %.2f\n", tr_end);
  fprintf(logfile,"Stream Processed in %.2f seconds\n\n",(tr_end -
tr_start));

  fprintf(rep,"\nEnded Executing this Stream at %s\n", ctime(&tim));
  fprintf(rep,"\nStream Started at %.2f\n", tr_start);
  fprintf(rep,"Stream Ended at %.2f\n", tr_end);
  fprintf(rep,"Stream Processed in %.2f seconds\n\n",
                (tr_end - tr_start));

  fprintf(logfile, "\nSQL statements processed: %d\n", stmt_cnt);
  /*fprintf(logfile, "Queries processed: %d\n", qry_cnt);*/

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}


/* SQLinit(): Perform initialization tasks.                    */
```

```
/*          Logs on to Oracle, opens some files and open a cursor for */
/*          later use.                                                */

void SQLinit() {

  int i;

  /* preallocate MAX_PREALLOC members of the dlist array
*/
  /* initializes others to NULL so that we can determine who to free later
*/

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc (sizeof(dltype));
      dlist[i]->defhdl = NULL;
/*      OCIhalloc(curq,&(dlist[i]->defhdl),OCI_HTYPE_DEFINE); */
    }
    else
      dlist[i] = NULL;
  }

  /* Connect to ORACLE.  Program will call sql_error()         */
  /* if an error occurs in connecting to the default database. */

  (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);

  if((status=OCIEnvInit((OCIEnv **)&tpcenv,OCI_DEFAULT,0,(dvoid
**)0)) !=
     OCI_SUCCESS)
    sql_error(tpcenv, status, 0);

  OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
  OCIhalloc(tpcenv,&curq,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cur_dml,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cur_ddl,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
  OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

  /* get username and password */

  passwd = strchr(logname, '/');
  *passwd = '\0';
  passwd++;

  if ((status = OCIServerAttach(tpcsrv,errhp,(text
*)0,0,OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER
,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(logname),OCI_
ATTR_USERNAME,
                errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_A
TTR_PASSWORD,
                errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                                        OCI_DEFAULT)) !=
OCI_SUCCESS)
    sql_error(errhp,status,1);

OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESSIO
N,errhp);

/*
  if ((status=OCILogon((OCIEnv *)tpcenv,(OCIError
*)errhp,(OCISvcCtx *)tpcsvc,
                        (text *)logname, strlen(logname), (text
*)passwd,
                        strlen(passwd), (text *) 0, 0)) !=
OCI_SUCCESS)
    sql_error(errhp, status, 1);
*/
  printf("\nConnected to ORACLE as user: %s\n\n", logname);

}



/* SQLexec()  Executes the SQL statement.                    */
/*          Parse the SQL statement.                         */
/*          If DDL or DML statements, execute right away.    */
/*          Else describe and define select list outputs,    */
/*              execute and fetch results.                   */

void SQLexec()
{
  int i;
  ub2 stmttyp = OCI_STMT_SELECT;     /* default is a SELECT
statement */

  /* Clause 5.3.6.2: QI(i,s) is the time between the first character */
  /*              of this query text is submitted and the first   */
  /*              character of the next query text is submitted.  */

  if (qry_cnt) {
    time(&tim);
    s_tr_end = gettime();
    fprintf(logfile,"Query Processed in %.2f seconds.\n\n",
     (s_tr_end - s_tr_start));

    /* print comments for this query that we have saved */

    /* fprintf(logfile, "%s\n", cmnt); */

    /*fprintf(rep, "Query %s : Execution time %.2f\n", qnp,(s_tr_end -
s_tr_start));*/
    fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
              qnp,(s_tr_end - s_tr_start),s_tr_start,s_tr_end);

    /* Let's fflush stuff so that we can see what's going on */

    /* Fix for Q15 */
    strcpy(qnp,qn);

    fflush(logfile);
    fflush(rep);
  }
  else
    tr_start = gettime();

  s_tr_start = gettime();

  /* prepare the statement */

  if ((status = OCIStmtPrepare(curq, errhp, (text*) stmt, (ub4)
strlen(stmt),
```

```
                                        OCI_NTV_SYNTAX,
OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);

  /* Prints the query text and comment to the logfile */

  fprintf(logfile, "\n%s\n", cmnt);
  cmnt[0]=0;
  fprintf(logfile, "\n%s\n", stmt);

  /* if this is a DDL or DML statement, execute it right away */
  /* only worries about SELECT statements right now, cannot   */
  /* execute a stored PL/SQL procedure in thie version        */


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_STM
T_TYPE,errhp);

  if (stmttyp != OCI_STMT_SELECT) {
    OCIsexec(tpcsvc,curq,errhp,1);
    return;
  }

  /* otherwise, this is a select statement */
  /* Describe and define output variables  */

  /* first let's execute it to get the select-list definition */

  OCIaset(curq, OCI_HTYPE_STMT, &pfmem, 0,
OCI_ATTR_PREFETCH_MEMORY, errhp);

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

  /* Need to get the number of rows fetched first   */
  /* since the following statments will screw it up */


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_RO
W_COUNT,errhp);

  /* To control memory usage, let's free up the extra dlist entries */
  /* that we have allocated.                                        */

  i=MAX_PREALLOC;
  while(dlist[i] != NULL) {
    free(dlist[i]);
    dlist[i++] = NULL;
  }

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);

  /* free all memory */

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (dlist[i] != NULL) {
      free(dlist[i]);
    }
  }

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


/* define_output_variables(): Describe and define select-list items for */
/*                            a query statement.                         */
/*                            Returns the number of select-list items    */
/*                            for this query.                            */

int define_output_variables()
{

  int i;
  int retflag = 0;

  for (i=0; i<MAX_SEL_LIST; i++) {

    slist[i].buflen = MAX_COLNAME_SIZE;

    if (OCIParamGet(curq, OCI_HTYPE_STMT, errhp, (dvoid **)
&tpcpar,
                                            POS(i)) != OCI_SUCCESS)
      break;


    /* dsize and nullok fields of dlist not used */

    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbsize),
                NULL, OCI_ATTR_DATA_SIZE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbtype),
                NULL, OCI_ATTR_DATA_TYPE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].buf),
                &(slist[i].buflen), OCI_ATTR_NAME, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].precision),
                NULL, OCI_ATTR_PRECISION, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].scale),
                NULL, OCI_ATTR_SCALE, errhp);

    /* For formatting purpose, remove trailing blanks in select-list name.
*/

/*
    if (slist[i].buflen < MAX_COLNAME_SIZE)
      (slist[i].buf)[slist[i].buflen] = '\0';
*/
    /* Well, we need to allocate for entries for dlist */

    if (i >= MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc(sizeof(dltype));
      dlist[i]->defhdl = NULL;
    }

    /* Let's check the sizes and types for this select list item */

    switch (slist[i].dbtype) {
```

```
   case OCI_TYPECODE_NUMBER:

     /* The odescr will not give a good estimate to the scale if */
     /* no scale was given in the Oracle table definition.       */

#ifdef HAVE_SCALE
     if (slist[i].scale != 0) {
               defbuf = (double *) dlist[i]->fbuf;
               deflen = FLT;
               deftype = OCI_TYPECODE_DOUBLE;
               slist[i].dbtype = OCI_TYPECODE_DOUBLE;
     } else {
               defbuf = (int *) dlist[i]->ibuf;
               deflen = INT;
               deftype = OCI_TYPECODE_INTEGER;
               slist[i].dbtype = OCI_TYPECODE_INTEGER;
     }
#else
     defbuf = (double *) dlist[i]->fbuf;
     deflen = FLT;
     deftype = OCI_TYPECODE_FLOAT;
     slist[i].dbtype = OCI_TYPECODE_FLOAT;
#endif /* HAVE_SCALE */

     break;

   default:

     /* default is character string */

     defbuf = (char **) dlist[i]->sbuf;
     deflen = MAX_STR_LEN;
     deftype = SQLT_STR;
/*     deftype = OCI_TYPECODE_CHAR; */
     break;
   }

   /* Define the column */

   if ((status=OCIDefineByPos(curq,&(dlist[i]->defhdl),errhp,POS(i),
                                          defbuf,deflen,deftype,NULL,
                                          dlist[i]-
>rlen,NULL,OCI_DEFAULT))!=OCI_SUCCESS)
     sql_error(errhp,status,1);
  }
  return i;
}


/* process_select_list(): Fetch rows from a query.                */

void process_select_list(num)
    int num;       /* number of select list items */
{

  int i,j;
  int ntf;
  int num_so_far;
  sword stats = OCI_SUCCESS;

  /* Print the headers for the query execution result */

  print_header(num);

  /* See if we need to limit the rows to fetch */

  ntf = (num_to_fetch >= 0) ? num_to_fetch : MAX_ARRAY;

  /* Fetch the rows and print them out */

  if ((ntf > MAX_ARRAY) || (num_to_fetch == -1)) {

    stats = OCIStmtFetch(curq, errhp, MAX_ARRAY,
OCI_FETCH_NEXT, OCI_DEFAULT);


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_RO
W_COUNT,errhp);

    print_rows(num,rows_ret);

    /* To avoid 1022 from OFEN */
    /* More rows to fetch...   */

    if (stats != OCI_NO_DATA) {
      if (num_to_fetch == -1) {
               while ((stats =
OCIStmtFetch(curq,errhp,MAX_ARRAY,OCI_FETCH_NEXT,
                                                    OCI_DEFAULT)) ==
OCI_SUCCESS) {
                 OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                              OCI_ATTR_ROW_COUNT,errhp);
                 print_rows(num,(num_so_far-rows_ret));
                 rows_ret = num_so_far;
               }
               /* Print the final rows */
               OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                            OCI_ATTR_ROW_COUNT,errhp);
               print_rows(num,(num_so_far-rows_ret));
               rows_ret = num_so_far;
      } else {
               ntf -= MAX_ARRAY;

               while ((stats = OCIStmtFetch(curq,errhp,
                                                    ((ntf>MAX_ARRAY) ?
MAX_ARRAY:ntf),
                                                    OCI_FETCH_NEXT,
OCI_DEFAULT)) ==
                        OCI_SUCCESS) {
                 ntf -= MAX_ARRAY;
                 OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                              OCI_ATTR_ROW_COUNT,errhp);
                 print_rows(num,(num_so_far-rows_ret));
                 rows_ret = num_so_far;
                 if (ntf <= 0) break;
               }
               OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                            OCI_ATTR_ROW_COUNT,errhp);
               print_rows(num,(num_so_far-rows_ret));
               rows_ret = num_so_far;
      }
    }
  } else {
    OCIStmtFetch(curq, errhp, ntf, OCI_FETCH_NEXT,
OCI_DEFAULT);

OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_RO
W_COUNT,errhp);
    print_rows(num,rows_ret);
  }

  fprintf(logfile,"\n\n%d %s processed.\n", rows_ret,
                rows_ret == 1 ? "row" : "rows");

}


int get_statement()
{
```

```
  char line[128];
  char *pos, *str;

  /* Reset statement buffer */

  stmt[0] = '\0';

  while (fgets(line, 127, qtemp) != NULL) {

    /* skip blank lines */
    if (line[0] == '\n')
      continue;

    /* remove blanks */

    str = line;

    while (*str == ' ') str++;

    /* Let's get the line together first */

    strcat(stmt, str);

    /* if this is a comment line */
    if ((str[0] == '-') && (str[1] == '-'))
      return COMMENT;

    /* see if this is a set_fetchrows line */
    if (strncmp(str, "set_fetchrows", 13) == 0) {
      pos = strchr(str, ';');
      *pos = '\0';
      pos = strchr(str, '=');
      num_to_fetch = atol(++pos);
      return SET_FETCHROW;
    }

    /* if this is the end of the current statement */
    if ((pos = strchr(stmt, ';')) != NULL) {
      *pos = '\0';
      return SQL_STMT;
    }
  }
  return END_OF_FILE;
}


/* memalloc(): Allocates memory, exit program if we have a problem. */

void *memalloc(size)
     int size;
{

  void *tmp;

  if ((tmp = (void *) malloc(size)) == NULL) {
    fprintf(stderr, "Error in malloc\n");
    SQLexit();
    return NULL;       /* should never reach here */
  } else {
    return tmp;
  }
}



void print_header(nsel)
     int nsel;          /* Number of select list items */
{

  int i, diff;
  char colname[MAX_COLNAME_SIZE];
  int len = 0;          /* Running column length */
  int cwid = 0;

  fprintf(logfile, "\n");

  for (i=0; i<nsel; i++) {

    /* extract the column name */

    strncpy((char *)colname, (char *)slist[i].buf, slist[i].buflen);
    colname[slist[i].buflen] = '\0';

    /* format the output a little */

    cwid = MAX(slist[i].dbsize, slist[i].buflen);

    /* do a little bit of formatting */

    if (cwid > 80) {
      fprintf(logfile,"\n");
      len = 0;
    } else if ((len += cwid) > 80) {
      fprintf(logfile,"\n");
      len = cwid;
    }
#ifdef FORMAT1
    if ((slist[i].dbtype == INT_TYPE) || (slist[i].dbtype == FLT_TYPE))
      fprintf(logfile, "%*s ", cwid, slist[i].buf);
    else  /* string type */
      fprintf(logfile, "%*s ", -cwid, slist[i].buf);
#else
      fprintf(logfile, "%*s ", -cwid, colname);
#endif /* FORMAT1 */
  }

  fprintf(logfile,"\n");
}


void print_rows(ncol, nrow)
     int ncol;
     int nrow;
{

  int i,j;
  int len;
  int diff;
  int cwid;

  for (i=0;i<nrow;i++) {

    len = 0;

    for (j=0;j<ncol;j++) {

      cwid = MAX(slist[j].dbsize, slist[j].buflen);

      /* do a little bit of formatting */

      if (cwid > 80) {
              fprintf(logfile,"\n");
              len = 0;
      } else if ((len += cwid) > 80) {
              fprintf(logfile,"\n");
              len = cwid;
      }

      switch(slist[j].dbtype) {
      case INT_TYPE:
```

```
#ifdef HAVE_SCALE
                fprintf(logfile, "%*ld|", cwid,       (dlist[j]->ibuf)[i]);
                break;
#endif /* HAVE_SCALE */
     case FLT_TYPE:
#ifdef FORMAT1
                fprintf(logfile,"%*.2f ", cwid, (dlist[j]->fbuf)[i]);
#else
                fprintf(logfile,"%*.2f ", -cwid, (dlist[j]->fbuf)[i]);
#endif /* FORMAT1 */
                break;
     default:
                fprintf(logfile, "%*s ", -(cwid), (dlist[j]->sbuf)[i]);
                break;
     }
    }
    fprintf(logfile, "\n");
  }
}


/* remove_newline(): Remove newline character from str. */

void remove_newline(str)
    char *str;
{

  char *p;

  while ((p = strchr(str,'\n')) != NULL)
    *p = ' ';
}
```

_____

qexecpl.h

_____

```
/*
 * $Header: qexecpl.h 13-nov-2001.17:52:35 mpoess Exp $
 */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.  */

/* NOTE:  See 'header_template.doc' in the 'doc' dve under the 'forms'
      directory for the header file template that includes instructions.
*/

/*
   NAME
     qexecpl.h

   DESCRIPTION
     SQL statement execution front-end header file.

   PUBLIC FUNCTION(S)
     <list of external functions declared/defined - with one-line
descriptions>

   PRIVATE FUNCTION(S)
     <list of static functions defined in .c file - with one-line descriptions>

   EXAMPLES

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   mpoess      11/13/01 - change DOP to 84 for DML and DDL
   mpoess      02/22/01 - add linux changes
   mpoess      08/05/99 - make compile
   mpoess      07/15/99 - Creation
   mpoess      07/15/99 - Creation

*/

/*
# ifndef S_ORACLE
#  include <s.h>
# endif
*/
#ifndef QSTREAMPL_H

#define QSTREAMPL_H

#include <stdio.h>
#include <string.h>
#include <sys/param.h>
#include <sys/types.h>
#include <time.h>
#include <errno.h>
#include <math.h>

#include <oratypes.h>

#include <oratypes.h>

#ifndef OCIDFN
#include <ocidfn.h>
#endif /* OCIDFN */

#ifndef OCI_ORACLE
#include <oci.h>
#endif /* OCI_ORACLE */
/*
#ifdef __STDC__
#include <ociapr.h>
#else
#include <ocikpr.h>
#endif *//* __STDC__ */

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

#ifndef TRUE
#define TRUE 1
#endif /* TRUE */

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */
#ifndef LINUX
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
#endif
/* defines and typedefs for parsing */

#define CRT_TBL  1
#define INS_STMT 3
#define SEL_STMT 4
#define UPD_STMT 5
#define DRP_VIEW 7
#define DRP_TBL  8
#define DEL_STMT 9
#define CRT_VIEW 10

/* defines and typedefs for query description */
```

```
#define MAX_COLNAME_SIZE 32   /* Maximum length of Column
name  */
#define MAX_SEL_LIST 16       /* Maximum items on a select list */

#define END_OF_LIST 1007      /* Error code when we reach the end of
the */
                      /* select list.                        */

/* types for describe */

#define CHAR_TYPE 1
#define NUM_TYPE  2
#define INT_TYPE  3
#define FLT_TYPE  4
#define STR_TYPE  5
#define DATE_TYPE 12

#define NUMWIDTH 16          /* Width of the numeric fields    */

#define POS(i) (i+1)         /* The position is 1...n instead  */
#define IND(i) (i-1)         /* of 0..n-1 as in an array.      */

typedef struct des
{
  ub2 dbsize;
  ub4 buflen;
/*  sb2 dsize;  */
  sb4 scale;
/*  sb2 nullok; */
  OCITypeCode dbtype;
/*  text buf[MAX_COLNAME_SIZE]; */
  text *buf;
  ub1 precision;
} sltype;


/* defines and typedefs for query select list definition */

#define MAX_ARRAY 50       /* Maximum array size for array fetch */
#define PFMEMSIZE 65536    /* Memory size of prefetch buffer     */

#define MAX_STR_LEN 256    /* Maximum size for string variables
*/
#define MAX_PREALLOC 8     /* Maximum number of preallocated
select list */
                      /* definitions.                        */

#define INT sizeof(long)
#define STR sizeof(char)
#define FLT sizeof(double)

#define FLTP (double *)
#define INTP (long *)
#define STRP (char **)

typedef struct def
{
  long ibuf[MAX_ARRAY];
  double fbuf[MAX_ARRAY];
  char sbuf[MAX_ARRAY][MAX_STR_LEN];
  ub2 rlen[MAX_ARRAY];         /* return length     */
  OCIDefine *defhdl;
} dltype;

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define NA              -1      /* ANSI SQL NULL */
#define VER7            2
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not
serializable */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define SID(sid) ((sid == -1) ? 0 : sid)

/* For get_statement */

#define END_OF_FILE -1
#define COMMENT 1
#define SQL_STMT   2
#define SET_FETCHROW 3


#define OCIhalloc(envh,hndl,htyp) \
   if((status=OCIHandleAlloc((dvoid *)envh,(dvoid
**)hndl,htyp,0,(dvoid **)0))!=OCI_SUCCESS) \
      sql_error(envh,status,0); \
   else \
      DISCARD 0

#define OCIhfree(hndl,htyp) \
   if((status=OCIHandleFree((dvoid *)hndl,htyp)) == OCI_SUCCESS) \
      fprintf(stderr, "Error freeing handle of type %d\n", htyp)

#define OCIaget(hndl,htyp,attp,size,atyp,errh) \
   if((status=OCIAttrGet((dvoid *)hndl,htyp,(dvoid *)attp,(dvoid
*)size,atyp,errh)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIaset(hndl,htyp,attp,size,atyp,errh) \
   if((status=OCIAttrSet((dvoid *)hndl,htyp,(dvoid
*)attp,size,atyp,errh)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0

#define OCIsexec(svch,stmh,errh,iter) \

if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DE
FAULT)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
      DISCARD 0


#define ISOTXT "alter session set isolation_level = serializable"
```

```
#define PDMLTXT "alter session force parallel dml parallel (degree
84)"
#define PDDLTXT "alter session force parallel ddl parallel (degree 84)"

#endif /* QSTREAMPL_H */
```

runTPCHall.afterload.AUDIT

```
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

sqlplus=$ORACLE_HOME/bin/sqlplus
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

if [ ! -f $RUN_ID_FILE ]
then
  echo "0" > $RUN_ID_FILE
fi

RUN_ID=`cat $RUN_ID_FILE`
#RUN_ID=`expr $RUN_ID + 1`
#echo $RUN_ID > $RUN_ID_FILE

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop
LD4IXCRE=${OUT_DIR}/Ld4ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
DAT_FILE=${KIT_DIR}/audit/1TB_final.dat

#echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID >
$SCRIPT_LOG_FILE
#echo >> $SCRIPT_LOG_FILE
#echo "Starting a new Oracle log file:
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >>
$SCRIPT_LOG_FILE
#echo >> $SCRIPT_LOG_FILE
#
#mv $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log.preAudit.$R
UN_ID
#touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
#
#echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
#bumpx.pl -s -o ${DAT_FILE} -p dbcre > $LD1DBCRE
#bumpx.pl -s -o ${DAT_FILE} -p sctso > $LD2SCTSO
#tshut
#2start
#STIME=`$GTIME`
#echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
#bumpx.pl -s -o ${DAT_FILE} -p dapop > $LD3DAPOP
#bumpx.pl -s -o ${DAT_FILE} -p ixcre > $LD4IXCRE
#bumpx.pl -s -o ${DAT_FILE} -p anlyz > $LD5ANLYZ
#2shut
#2start
#ckpnt.sh
#echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE

2start
ckpnt.sh

echo "Start: dbtables.sql and count.sql" >> $SCRIPT_LOG_FILE
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/dbtables >
${RDB_TABLES} 2>&1
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/firstten >
${FIRST_TEN} 2>&1
echo "End: dbtables.sql and count.sql `date`" >> $SCRIPT_LOG_FILE

#2shut >>  $SCRIPT_LOG_FILE
#2start >>  $SCRIPT_LOG_FILE
#ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

2shut >>  $SCRIPT_LOG_FILE
#sleep 300
#2start >>  $SCRIPT_LOG_FILE
#ckpnt.sh
#runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

#sleep 600
#2shut >> $SCRIPT_LOG_FILE

#cp $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$OUT_DIR

#echo "End TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID
`date`" >> $SCRIPT_LOG_FILE
```

runTPCHall.beforeload.AUDIT

```
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

sqlplus=$ORACLE_HOME/bin/sqlplus
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

if [ ! -f $RUN_ID_FILE ]
then
  echo "0" > $RUN_ID_FILE
fi

RUN_ID=`cat $RUN_ID_FILE`
#RUN_ID=`expr $RUN_ID + 1`
echo $RUN_ID > $RUN_ID_FILE

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
```

```
LD3DAPOP=${OUT_DIR}/Ld3dapop
LD4IXCRE=${OUT_DIR}/Ld4ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
DAT_FILE=${KIT_DIR}/audit/3TB.dat

echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID >
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file:
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

mv $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log.preAudit.$R
UN_ID
touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log

echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -o ${DAT_FILE} -p dbcre > $LD1DBCRE
bumpx.pl -s -o ${DAT_FILE} -p sctso > $LD2SCTSO
tshut
sleep 300
2start
stime=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -o ${DAT_FILE} -p dapop > $LD3DAPOP
bumpx.pl -s -o ${DAT_FILE} -p ixcre > $LD4IXCRE
bumpx.pl -s -o ${DAT_FILE} -p anlyz > $LD5ANLYZ
2shut
sleep 300
2start
ckpnt.sh
2shut
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo Generated seed: `cat $KIT_DIR/audit/seed` >>
$SCRIPT_LOG_FILE

exit
```

runTPCHpt.AUDIT

```
#!/bin/ksh
. $KIT_DIR/env
#set -x
#ECHO=/bin/echo
SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of the query
template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

UPD_SQL=${UPD_DIR}/sql
UPD_SPT=${UPD_DIR}/scripts
UPD_SRC=${UPD_DIR}/source
UPD_DAT=${UPD_DIR}/data

TPCD_BIN=${KIT_DIR}/audit/bin

GTIME=${SRC_DIR}/gtime
SEED_FILE=${KIT_DIR}/audit/seed

DF=/dev/null
HID=1
INTERVAL=60
COUNT=1200

# The defaults

QPROG=${QEXEC}/qexec

usage () {

echo " "
echo "Usage: $0 [-p <program for query stream>] [-u1 <program for
UF1>]"
echo "         [-u2 <program for UF2>] [-o] [-s] [-h] [-u
<user/password>]"
echo "         <scale factor> <run_number>"
echo ""
echo "scale factor     : The scale factor of the run."
echo "update ||ism     : The parallelism to use for the UFs."
echo ""
echo "-p <program>     : Program for Query Stream."
echo "                   Default is $QPROG."
echo "-u1 <program>    : Program for UF1."
echo "                   Default is $U1PROG."
echo "-u2 <program>    : Program for UF2."
echo "                   Default is $U2PROG."
echo "-o               : Collect Oracle statistics."
echo "-s               : Collect System statistics."
echo "-u <user/passwd> : User/Password. Default is tpch/tpch."
echo "-h               : Displays this message."
}
set -- `getopt "p:u1:u2:osu:h" "$@"` || usage

while :
do
  case "$1" in
  -u1) shift; U1PROG=$1;;
  -u2) shift; U2PROG=$1;;
  -p) shift; QPROG=$1;;
  -o) OSTAT=1;;
  -s) SSTAT=1;;
  -h) usage; exit 0;;
  --) shift; break;;
  esac
  shift;
done


if [ "$#" -ne "3" ]
then
  usage
  exit 1
fi

SF=$1
PARA=$2
RUN_ID=$3

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
        mkdir $OUT_DIR
fi

TPCD_LOG=${OUT_DIR}
TPCD_RPT=${OUT_DIR}
```

```
OUT=${OUT_DIR}

let UF_SET="($PARA-1)*($NUM_STREAMS+1)+1"
START_SET=1
let STOP_SET=$NUM_STREAMS
let START_SET_UPDATE="($PARA-1)*($NUM_STREAMS+1)+2"
let
STOP_SET_UPDATE="$START_SET_UPDATE+$NUM_STREAMS
-1"

TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s0
TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s0inter
QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s0
QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.0
SCRIPT_LOG_FILE=${TPCD_LOG}/m${PARA}timing
UF1_LOG=${TPCD_LOG}/m${PARA}s0rf1
UF2_LOG=${TPCD_LOG}/m${PARA}s0rf2
STREAM_COUNT_LOG=${TPCD_LOG}/m${PARA}tstrcnt

echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID} `date`"
> $SCRIPT_LOG_FILE
echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID} `date`"
> $TPCD_RPT_FILE
echo "Generates query template file with seed: `cat $SEED_FILE` for
stream 0" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

${QGEN} -c -r `cat $SEED_FILE` -p 0 -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}

START=`$GTIME`
echo "Start Power Test - RUN:${PARA} SEQUENCE:${RUN_ID}
Execution Starts $START, `date`" >>  $SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

# Execute UF1

SDATE=`date`
UF1_START=`$GTIME`
echo "Start UF1  $UF1_START, `date`" >> $SCRIPT_LOG_FILE

${UPD_SPT}/runuf1.sh ${UF_SET} >> $UF1_LOG 2>&1
# Execute Query Stream

UF1_END=`${GTIME}`
E1DATE=`date`

UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo UF1: Execution Time: $UF1_TIME >> ${TPCD_RPT_FILE}
echo Start Time: $UF1_START, $SDATE >> ${TPCD_RPT_FILE}
echo End Time: $UF1_END, $E1DATE >> ${TPCD_RPT_FILE}
echo "" >> ${TPCD_RPT_FILE}

echo "End UF1 $UF1_END, ${E1DATE}" >> $SCRIPT_LOG_FILE
echo UF1: Execution Time: $UF1_TIME >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

echo "Start Query Part `$GTIME`, `date` " >> $SCRIPT_LOG_FILE

${QPROG} ${DATABASE_USER} q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE}  > $DF  2>&1

# Execute UF2

UF2_START=`${GTIME}`
E2DATE=`date`

echo "End Query Part `$GTIME`, ${E2DATE}" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

echo "Start UF2  $UF2_START, `date`" >> $SCRIPT_LOG_FILE
${UPD_SPT}/runuf2.sh ${UF_SET} >> $UF2_LOG 2>&1
UF2_END=`${GTIME}`
END=`${GTIME}`
EDATE=`date`

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
echo UF2: Execution Time: $UF2_TIME >> ${TPCD_RPT_FILE}
echo Start Time: $UF2_START, $E2DATE >> ${TPCD_RPT_FILE}
echo End Time: $UF2_END, $EDATE >> ${TPCD_RPT_FILE}


echo "End UF2 $UF2_END, $EDATE" >>  $SCRIPT_LOG_FILE
echo UF2: Execution Time: $UF2_TIME >> $SCRIPT_LOG_FILE
echo >>  $SCRIPT_LOG_FILE

echo "End TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID}, $END, $EDATE" >> $SCRIPT_LOG_FILE
MEA_INT=`echo $END - $START | bc`
echo "Elapsed Time for TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID} is $MEA_INT" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
i=$START_SET
PSEED=`cat $SEED_FILE`

while [ $i -le $STOP_SET ]; do
  TPCD_LOG_FILE=${TPCD_LOG}/mt${RUN_ID}_${i}.log
  TPCD_RPT_FILE=${TPCD_RPT}/mt${RUN_ID}_${i}.rpt
            QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.${i}
  QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}

  PSEED=`expr $PSEED + 1`
  ${QGEN} -c -r ${PSEED} -p ${i} -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}

  i=`expr $i + 1`
done

TH_START_D=`date`
TH_START_T=`${GTIME}`
echo >> $SCRIPT_LOG_FILE

rm -f /tmp/th_pipe1
mknod /tmp/th_pipe1 p
rm -f /tmp/th_pipe2
mknod /tmp/th_pipe2 p
i=$START_SET

echo "Start Throughput Test - RUN:${PARA}
SEQUENCE:${RUN_ID} $TH_START_T, $TH_START_D"  >>
$SCRIPT_LOG_FILE

# starts a script to count the streams during the throughtput run
(scnt.sh $PARA $RUN_ID > $STREAM_COUNT_LOG &)

while [ $i -le $STOP_SET ]; do
  M_SDATE=`date`
  M_STIME=`${GTIME}`
            TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
            TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s${i}inter
  echo "Start Query Stream $i $M_STIME, ${M_SDATE}" >>
$SCRIPT_LOG_FILE
            QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}
  ${QPROG} ${DATABASE_USER} q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE} | grep -v "Connected to
ORACLE" >> $SCRIPT_LOG_FILE &
  i=`expr $i + 1`
```

```
done

(${KIT_DIR}/audit/runTPCHus $RUN_ID $START_SET_UPDATE
$STOP_SET_UPDATE ${SF} $PARA >> $SCRIPT_LOG_FILE 2>&1
&)

wait
THQ_END_T=`$GTIME`
THQ_END_D=`date`
echo End all Query Streams $THQ_END_T, $THQ_END_D >>
$SCRIPT_LOG_FILE
print > /tmp/th_pipe1
read < /tmp/th_pipe2

TH_END_D=`date`
TH_END_T=`$GTIME`
echo End Update Stream ${TH_END_T}, ${TH_END_D} >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "End Throughput Test ${TH_END_T}, ${TH_END_D}" >>
$SCRIPT_LOG_FILE
echo Execution Time Throughput Test: `echo ${TH_END_T} -
${TH_START_T} | bc` >> $SCRIPT_LOG_FILE

i=$START_SET
while [ $i -le $STOP_SET ]; do
        TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
 ${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
        i=`expr $i + 1`
done
PIDS=`ps -fu oracle | grep scnt.sh | grep -v grep | awk '{print $2}'`
kill -9 $PIDS
#calculate the metric
#analyze_streams.pl -f p -n $RUN_ID >
${TPCD_RPT}/tpch_metric.${RUN_ID}.${HID}.rpt
```

runTPCHus

```
#!/bin/ksh
. $KIT_DIR/env

SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
UPD_SPT=${UPD_DIR}/scripts
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of the query
template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

RUN_ID=$1
START_SET_UPDATE=$2
STOP_SET_UPDATE=$3
SF=$4
PARA=$5

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
        mkdir $OUT_DIR
fi


TPCD_RPT=$OUT_DIR
SCRIPT_LOG_FILE=${OUT_DIR}/m${PARA}timing
OUT=$OUT_DIR

GTIME=${SRC_DIR}/gtime
HID=1

START=`$GTIME`
echo "Start Update Stream $START, `date`" >> $SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE


#waiting for all the query streams to finish first
read < /tmp/th_pipe1

i=$START_SET_UPDATE
j=1
while [ $i -le $STOP_SET_UPDATE ]; do

 # Execute UF1

 UF1_LOG=${OUT_DIR}/m${PARA}s${j}rf1
 UF2_LOG=${OUT_DIR}/m${PARA}s${j}rf2
 RPT_FILE=${OUT_DIR}/m${PARA}s${j}inter

 SDATE=`date`
 UF1_START=`$GTIME`
        echo "Start UF1-${j} at ${UF1_START}, ${SDATE}" >>
${RPT_FILE}

 ${UPD_SPT}/runuf1.sh ${i} >> ${UF1_LOG} 2>&1
 UF1_END=`${GTIME}`
        EDATE=`date`
 echo "End UF1-${j} at ${UF1_END}, ${EDATE}" >> ${RPT_FILE}
        echo UF1-${j} Execution Time: `echo ${UF1_END} -
${UF1_START} | bc` >> ${RPT_FILE}

 # Execute UF2

 SDATE=`date`
 UF2_START=`${GTIME}`
 echo "Start UF2-${j} ${UF2_START}, ${SDATE}" >>
${RPT_FILE}

 ${UPD_SPT}/runuf2.sh ${i} >> ${UF2_LOG} 2>&1
 UF2_END=`${GTIME}`
 EDATE=`date`
 echo "End UF2-${j} at $UF2_END, ${EDATE}" >> ${RPT_FILE}
        echo UF2-${j} Execution Time: `echo ${UF2_END} -
${UF2_START} | bc` >> ${RPT_FILE}

 i=`expr $i + 1`
        j=`expr $j + 1`
done

print > /tmp/th_pipe2
```

runuf1.sh

```
#/te!/bin/ksh
#
# $Header: runuf1.sh 25-oct-2001.15:56:04 mpoess Exp $
#
# runuf1.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#    NAME
```

```
#     runuf1.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
#             -p [<program>] <run_id> <scale factor> <pair number>
#             <parallelism>
# USAGE
#    To execute UF1.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      10/25/01 - change default directory for update sets
#    mpoess      10/17/01 - add support for external tables
#    mpoess      08/15/99 - Creation
#    mpoess      08/15/99 - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
#PAR_HINT=${UPDATE_DOP}
PAR_HINT=32

LOGPATH=.
PASSWD=${DATABASE_USER}
if [ $# -lt 1 ];
then
  echo runuf1.sh setnum
  exit 1
fi
SETNUM=$1
i=1
PID=""

# perform the update function 1

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as '/home/oracle/OraHome1/dbs/flatfiles';

drop table temp_l_et;
create table temp_l_et(
   l_orderkey          number ,
   l_partkey           number ,
   l_suppkey           number ,
   l_linenumber        number ,
   l_quantity          number ,
   l_extendedprice     number ,
   l_discount          number ,
   l_tax               number ,
   l_returnflag        char(1) ,
   l_linestatus        char(1) ,
   l_shipdate          date ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipinstruct      char(25) ,
   l_shipmode          char(10) ,
   l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
            records delimited by newline
            badfile 'l_et.${SETNUM}.bad'
            logfile 'l_et.${SETNUM}.log'
            fields terminated by '|'
            missing field values are null
      )
      location (
      'lineitem.tbl.u${SETNUM}'
))
reject limit unlimited;

drop table temp_o_et;
create table temp_o_et(
   o_orderkey          number ,
   o_custkey           number ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_orderdate         date ,
   o_orderpriority     char(15) ,
   o_clerk             char(15) ,
   o_shippriority      number ,
   o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
            records delimited by newline
            badfile 'o_et.${SETNUM}.bad'
            logfile 'o_et.${SETNUM}.log'
            fields terminated by '|'
            missing field values are null
      )
      location (
      'orders.tbl.u${SETNUM}'
))
reject limit unlimited;
alter table temp_l_et parallel ${PAR_HINT};
alter table temp_o_et parallel ${PAR_HINT};

alter session force parallel dml parallel (degree ${PAR_HINT});
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj = 10;
insert into orders (
  select
   o_orderdate        ,
   o_orderkey         ,
   o_custkey          ,
   o_orderpriority    ,
   o_shippriority     ,
   o_clerk            ,
   o_orderstatus      ,
   o_totalprice       ,
   o_comment
from temp_o_et);

insert into lineitem (
  select
```

```
   l_shipdate      ,
   l_orderkey      ,
   l_discount      ,
   l_extendedprice ,
   l_suppkey       ,
   l_quantity      ,
   l_returnflag    ,
   l_partkey       ,
   l_linestatus    ,
   l_tax           ,
   l_commitdate    ,
   l_receiptdate   ,
   l_shipmode      ,
   l_linenumber    ,
   l_shipinstruct  ,
   l_comment
from temp_l_et);
commit;

drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`

# Done

echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#   NAME
#     runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#   runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#             <scale factor> <pair number> <parallelism>
# USAGE
#   To execute UF2.
#
#   NOTES
#     <other useful comments, qualifications, etc.>
#
#   MODIFIED   (MM/DD/YY)
#   mpoess     10/25/01 - change default directory for update sets
#   mpoess     10/17/01 - add support for external tables
#   mpoess     08/15/99 - Creation
#   mpoess     08/15/99 - Creation
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
#PAR_HINT=${UPDATE_DOP}
PAR_HINT=128
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

if [ $# -lt 1 ]
then
  usage
  exit 1
fi

SETNUM=$1

i=1
PID=""

START=`$GTIME`
# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing  on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as '/home/oracle/OraHome1/dbs/flatfiles';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
   t_orderkey          number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
            records delimited by newline
            badfile 'okey.${SETNUM}.bad'
            logfile 'okey.${SETNUM}.log'
            fields terminated by '|'
            missing field values are null
      )
      location (
      'delete.${SETNUM}'))
reject limit unlimited;

rem alter table temp_okey_et parallel ${PAR_HINT};
alter table temp_okey_et parallel 32;

rem create table temp_okey parallel ${PAR_HINT} nologging as select
* from temp_okey_et;
create table temp_okey parallel 32 nologging as select * from
temp_okey_et;

rem create unique index i_temp_okey on temp_okey (t_orderkey)
parallel ${PAR_HINT} nologging compute statistics;
create unique index i_temp_okey on temp_okey (t_orderkey) parallel 32
nologging compute statistics;

analyze table temp_okey estimate statistics sample 2 percent;

alter session  force parallel dml parallel ${PAR_HINT};
alter session set isolation_level=serializable;
alter session set optimizer_index_cost_adj = 10;
```

```
delete from (select /*+ use_nl(o) */ o.rowid from orders o, temp_okey t
where o.o_orderkey = t.t_orderkey order by 1);

delete from (select /*+ use_nl(l) */ l.rowid from lineitem l,temp_okey t
where l.l_orderkey = t.t_orderkey order by 1);

commit;

drop table temp_okey;
drop table temp_okey_et;
exit;
!
```

```
END=`$GTIME`

# Done

echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

Appendix G: Price Quotes

January 29, 2004

Raghunath K. Othayoth
ISS - Solutions and Strategy
Hewlett-Packard Company
281-518-2748 tel

Per your request for information on pricing Red Hat Enterprise Linux to be used in conjunction with your TPC-H benchmark testing, please find the quote below. These prices are valid for 30 days.

Part Number	***Description***	***Unit Price***	***Quantity***	***Price***
SVC0022	Red Hat Enterprise Linux AS (version 3 Standard Edition for x86) subscription	$1,499	8	$11,992
SVC0022	2 Additional Years Subscription to Red Hat Enterprise Linux AS (version 3 Standard Edition for x86)	$1,499	16	$23,984
TOTAL				$35,976.00

Products are orderable through www.redhat.com or Red Hat Sales 1-888-REDHAT-1. If we can be of any further assistance, please contact Mike Ferris at mferris@redhat.com.

*Support and maintenance for software includes annual phone and web support and continuous proactive update and upgrade support via Red Hat Network.

-----Original Message-----
From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Tuesday, March 02, 2004 3:47 PM
To: Othayoth, Raghunath
Subject: Pricing Request

Product	**Price**	**Quantity**	**Extended Price**
Oracle Database 10g, Enterprise Edition, Named User Plus for 3 years	$10,000	32	$320,000
Real Application Clusters, Named User Plus for 3 years	$5,000	32	$160,000
Partitioning, Named User Plus for 3 years	$2,500	32	$80,000
Database Server Support Package (**8** servers, 3 years)	$16,000	3 years	$48,000
Oracle Mandatory E-Business Discount			<$121,600>
Total			$486,400

Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081