Hewlett-Packard Company

TPC Benchmark™ H
Full Disclosure Report

HP BladeSystem c-Class
ProLiant BL460c IB Cluster 16P DC

using
Oracle Database 10g Release 2, Enterprise Edition with
Real Application Cluster and
Partitioning; and
Red Hat Enterprise Linux 4

First Edition
August 2007

First Edition – August 2007

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.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on the HP BladeSystem c-Class ProLiant BL460c IB Cluster 16P DC 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 2.6.1. The operating system used for the benchmark was Red Hat Enterprise Linux 4.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	QppH @ 300GB	QthH @ 300GB	QphH @ 300GB	$ / QphH @ 300GB
HP BladeSystem c-Class ProLiant BL460c IB Cluster 16P DC consisted 8 x ProLiant BL460c	Oracle Database 10g Enterprise Edition R2 with Real Application Cluster and Partitioning, and Red Hat Enterprise Linux 4	$497,869 USD	49,541.3	31675.3	39,613.6	$12.57 USD
16 x HP ProLiant BL465c with HP StorageWorks SB40c	Red Hat Enterprise Linux 5					

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 BladeSystem c-Class ProLiant BL460c IB Cluster 16P DC can be found in the following pages.

Auditor

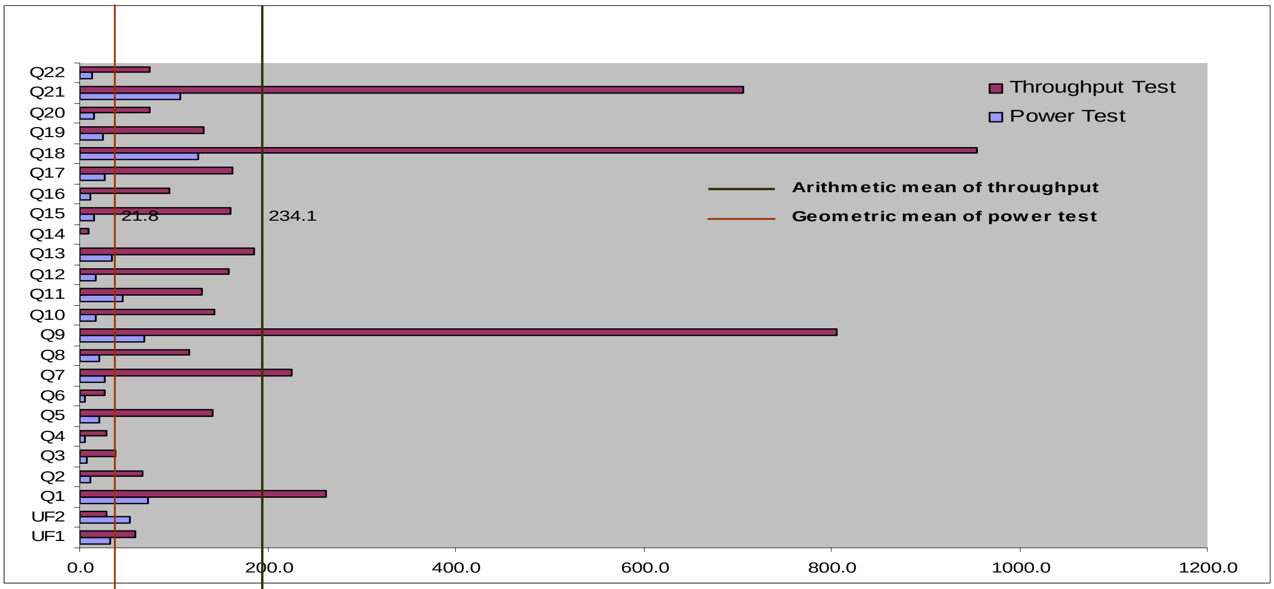
The benchmark configuration, environment and methodology were audited by Lorna Livingtree of Performance Metrics Inc. to verify compliance with the relevant TPC specifications.

Table Of Contents

	HP BladeSystem c-Class **ProLiant BL460c IB Cluster 16P DC**	TPC-H Rev. 2.6.1 Report Date: **August 9, 2007**

Total System Cost	Composite Query per Hour Metric	Price / Performance
$497,869USD	**39,613.6** QphH@300GB	**$12.57 USD** $ / QphH@300GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
300GB	**Oracle Database 10g Release 2, Enterprise Edition with Oracle Real Application Clusters and Partitioning**	**Red Hat Enterprise Linux 4**		**September 15, 2007**



Database Load Time = 1:29:24	Load Included Backup: N	Total Data Storage / Database Size = 17.28
RAID (Base tables only): Y	RAID (Base tables and auxiliary data structures): Y	RAID (All): Y

System Components	Total	Per Node
HP BladeSystem c7000 Enclosure:	1	n.a.
HP ProLiant BL460c (Database Server):	8	n.a
Processors (Dual-Core Intel Xeon X5160, 3.0 GHz, 1333 MHz FSB)/Cores/Threads:	16/32/32	2/4/4
Memory:	128GB	16GB
HP 36GB 10K SAS 2.5 Hot Plug HDD (OS)	16	2
On board Dual-Port NIC:	8	1
HP GbE2 Switch:	1	n.a.
HP 4X DDR InfiniBand Mezzanine HCA:	16	2
HP 4X DDR InfiniBand Switch (Oracle Cluster Interconnect):	2	n.a
Storage Components		
HP BladeSystem c7000 Enclosure:	2	
HP 4X DDR InfiniBand Switch (Storage Interconnect):	2	
HP ProLiant BL465c (1 x Dual-Core AMD Opteron 2214 HE. 2.2 GHz, 2GB Main Memory, 2x HP 36GB 15k 2.5 SAS drive, 1 x HP 4X DDR InfiniBand Mezzanine HCA (Storage Server)	16	
HP StorageWorks SB40c with 6 x HP 36GB 15k 2.5 SAS drive (Storage Blade):	16	
Total Storage:	5184 GB	

HP BladeSystem c-Class
ProLiant BL460c IB Cluster 16P DC

TPC-H Rev. 2.6.1

Report Date: 9-Aug-07

Description	Part Number	Brand	Pricing	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware							
HP ProLiant BL460c CTO Blade	404667-B21		1	1462	8	11696	
Dual-Core Intel Xeon Processor 5160 (3.00 GHz, 1333 MHz FSB) Option Kit	416673-L21		1	1499	16	23984	
HP 4GB FBD PC2-5300 2x2GB Kit	397413-B21		1	749	32	23968	
HP 4X DDR InfiniBand Mezzanine HCA for HP BladeSystem c-Class	410533-B21		1	579	16	9264	
HP 36GB 10K SAS 2.5 Hot Plug Hard Drive	375859-B21		1	269	16	4304	
HP 3y 4h 24x7 c-Class Svr Blade HW Supp	UE459E		1	369	8		2952
HP BLc7000 3 PH 6 Fan NA/Jp Trl ICDC Kit	412133-B21		1	6699	1	6699	
HP BLc Bnt 1GbE2 Switch Opt Kit	410917-B21		1	1399	1	1399	
HP BLc 4X DDR InfiniBand Switch Option Kit	410398-B21		1	5999	2	11998	
HP 3y 4h 24x7 c7000 Enclosure HW Supp	UE479E		1	927	1		927
HP 5642 Pallet Unassembled Rack	358254-B21		1	689	1	689	
HP CAT5 KVM USB 1 Pack Interface Adapter	336047-B21		1	99	1	99	
HP s7540 CRT Monitor	PF997AA#ABA		1	139	1	139	
HP USB Standard Keyboard	DX752AV#ABA		1	12	1	12	
HP USB 2-Button Optical Scroll Mouse	PT951AV		1	5	1	5	
					Subtotal	**94,256**	**3,879**
Storage							
HP ProLiant BL465c CTO Blade	403435-B21		1	1362	16	21792	
AMD Opteron processor Model 2210 (1.8 GHz, 95W) Processor Option Kit, FIO Base	414210-L21		1	249	16	3984	
HP 1GB Reg PC2-5300 2x512M Kit	408850-B21		1	189	32	6048	
HP 4X DDR InfiniBand Mezzanine HCA for HP BladeSystem c-Class	410533-B21		1	579	16	9264	
HP 3y 4h 24x7 c-Class Svr Blade HW Supp	UE459E		1	369	16		5904
HP BLc SB40c Storage Blade	411243-B21		1	1599	16	25584	
HP 36GB 15k 2.5 single port SAS drive	431933-B21		1	369	128	47232	
HP 3y 4h 24x7 BLcSB40c StrgBlade HW Supp	UF207E		1	343	16		5488
HP BLc7000 3 PH 6 Fan NA/Jp Trl ICDC Kit	412133-B21		1	6699	2	13398	
RHEL 5 Media Only SW	452561-B21		1	50	1	50	
HP 3y 24x7 RH 8pack SW Support	UH226E		1	6718	2		13,436
HP BLc7000 3 PH 6 Fan NA/Jp Trl ICDC Kit	412133-B21		1	6699	2	13398	
HP 3y 4h 24x7 c7000 Enclosure HW Supp	UE479E		1	927	2		927
HP 2M 4X DDR IB Copper Cable	410123-B23		1	68	6	408	
					Subtotal	**141,158**	**25,755**
Software							
Oracle Database 10g Enterprise Edition Relase 2, Named User Plus for 3 Years	run-time	Oracle	2	10000	16 **	160,000	
Oracle Real Application Clusters, Named User Plus for 3 Years	run-time	Oracle	2	5000	16 **	80,000	
Partitioning, Named User Plus for 3 Years	run-time	Oracle	2	2500	16 **	40,000	
Database Server Support Package for 3 Years	run-time	Oracle	2	6000	8		48,000
RHEL 4 AS Prm 3YR 24x7	384958-B21		1	1909	8	15,272	Included
					Subtotal	**295,272**	**48,000**
HP Large Purchase and Net30 discount			1			(40,110)	(4,741)
Oracle Mandatory E-Business Discount on Licenses and Support		Oracle	2			(65,600)	
					Total	**424,976**	**72,893**

3-Year Cost of Ownership in USD:	**$497,869 USD**
QphH Rating:	**39,613.6**
USD$ / QphH@300GB:	**$12.57 USD**

Pricing: 1- HP Direct at 16% discount 800-203-7648
Pricing: 2- Oracle pricing contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081
(**) 16 = 32 x 0.50. Explanation: for the purposes of counting the number of processors which require licensing for Intel multicorechips, "n" cores shall be determined by multiplying the total number of cores by a core processor licensing factor of .50. Oracle Price quote is in Appendix G.

All discounts are based on US list prices and for similar quantities and configurations. 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 section of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you

Numerical Quantities

Measurement Results:

Database Scale Factor	= 300
Total Data Storage / Database Size	= 17.28
Start of Database Load	= 8/3/2007 14:34:44
End of Database Load	= 8/3/2007 16:04:08
Database Load Time	= 1:29:24
Query Streams for Throughput Test	= 9
TPC-H Power	= 49,541.3
TPC-H Throughput	= 31,675.3
TPC-H Composite Query-per-Hour Metric (QphH@300GB)	= 39,613.6
Total System Price Over 3 Years	= $497,869USD
TPC-H Price/ Performance Metric ($/QphH@300GB)	= $12.57USD

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 6751.0 seconds

Duration of Stream Execution:

Power Stream	Seed	RF1 Start Time	Query Start Time	RF2 Start Time	Duration
		RF1 End Time	Query End Time	RF2 End Time	
	803160408	8/3/2007 16:17:52	8/3/2007 16:18:46	8/3/2007 16:30:28	0:13:09
		8/3/2007 16:18:46	8/3/2007 16:30:28	8/3/2007 16:31:01	

Throughput Stream	Seed	Query Start Time	Duration	RF1 Start Time	RF2 Start Time
		Query End Time		RF1 End Time	RF2 End Time
1	803160409	8/3/2007 16:31:09	1:38:10	8/3/2007 18:10:27	8/3/2007 18:11:05
		8/3/2007 18:09:19		8/3/2007 18:11:05	8/3/2007 18:11:37
2	803160410	8/3/2007 16:31:09	1:34:15	3/5/2007 18:11:37	8/3/2007 18:12:26
		8/3/2007 18:05:24		8/3/2007 18:12:26	8/3/2007 18:12:54
3	803160411	8/3/2007 16:31:09	1:39:18	8/3/2007 18:12:54	8/3/2007 18:13:43
		8/3/2007 18:10:27		8/3/2007 18:13:43	8/3/2007 18:14:14
4	803160412	8/3/2007 16:31:09	0:58:01	8/3/2007 18:14:14	8/3/2007 18:15:16
		8/3/2007 17:29:10		8/3/2007 18:15:16	8/3/2007 18:15:46
5	803160413	8/3/2007 16:31:09	1:35:25	8/3/2007 18:15:46	8/3/2007 18:16:58
		8/3/2007 18:06:34		8/3/2007 18:16:58	8/3/2007 18:17:26
6	803160414	8/3/2007 16:31:09	1:31:43	8/3/2007 18:17:26	8/3/2007 18:18:38
		8/3/2007 18:02:52		8/3/2007 18:18:38	8/3/2007 18:19:05
7	803160415	8/3/2007 16:31:09	1:26:14	8/3/2007 18:19:05	8/3/2007 18:20:12
		8/3/2007 17:57:23		8/3/2007 18:20:12	8/3/2007 18:20:42
8	803160416	8/3/2007 16:31:09	1:16:37	8/3/2007 18:20:42	8/3/2007 18:21:51
		8/3/2007 17:47:46		8/3/2007 18:21:51	8/3/2007 18:22:19
9	803160417	8/3/2007 16:31:09	1:12:48	8/3/2007 18:22:19	8/3/2007 18:23:18
		8/3/2007 17:43:57		8/3/2007 18:23:18	8/3/2007 18:23:40

TPC-H Timing Intervals (in seconds)

Query	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Stream00	72.0	12.2	7.8	5.6	20.5	5.4	26.0	21.9
Stream01	89.2	127.1	104.3	10.0	353.5	19.7	341.5	114.2
Stream02	314.3	44.4	37.6	15.1	89.6	21.2	87.0	85.4
Stream03	635.9	121.5	6.5	22.7	109.2	19.0	507.6	131.2
Stream04	127.9	39.9	30.9	17.9	104.4	77.9	129.5	57.9
Stream05	121.8	26.0	26.9	54.9	92.7	47.5	341.2	84.3
Stream06	318.0	53.0	77.4	88.6	183.6	27.0	324.4	98.5
Stream07	336.5	77.6	27.9	34.0	64.0	16.9	291.2	146.5
Stream08	482.5	73.0	23.2	17.4	179.0	17.8	103.8	293.1
Stream09	120.3	104.2	47.7	17.6	215.6	18.3	101.5	141.1
Min Qi	72.0	12.2	6.5	5.6	20.5	5.4	26.0	21.9
Max Qi	635.9	127.1	104.3	88.6	353.5	77.9	507.6	293.1
Avg Qi	261.8	67.9	39.0	28.4	141.2	27.1	225.4	117.4
Query	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**	**Q15**	**Q16**
Stream00	69.1	16.7	46.4	17.7	35.3	2.5	16.2	12.4
Stream01	368.1	212.5	208.4	308.6	119.1	6.2	170.4	194.6
Stream02	991.0	481.3	119.3	105.6	496.8	10.4	104.0	168.4
Stream03	1309.2	267.9	80.6	27.7	104.7	10.3	78.7	11.6
Stream04	516.5	86.1	132.2	271.0	169.2	7.5	337.7	51.0
Stream05	454.3	66.6	98.2	123.8	230.5	7.8	406.3	255.7
Stream06	873.7	72.0	203.6	336.0	243.8	7.2	140.6	43.5
Stream07	563.1	96.4	177.5	147.0	143.6	10.3	70.8	53.9
Stream08	1095.9	79.2	120.1	199.6	112.3	10.8	204.5	52.3
Stream09	1825.1	52.0	116.9	48.9	210.4	13.5	81.9	114.0
Min Qi	69.1	16.7	46.4	17.7	35.3	2.5	16.2	11.6
Max Qi	1825.1	481.3	208.4	336.0	496.8	13.5	406.3	255.7
Avg Qi	806.6	143.1	130.3	158.6	186.6	8.7	161.1	95.7
Query	**Q17**	**Q18**	**Q19**	**Q20**	**Q21**	**Q22**	**RF1**	**RF2**
Stream00	25.9	127.1	24.4	15.3	106.9	14.2	53.6	33.0
Stream01	132.1	1057.1	144.7	176.9	1598.0	34.1	38.4	31.2
Stream02	426.2	1513.8	187.9	48.0	232.1	75.9	49.2	27.7
Stream03	175.5	1465.1	55.3	63.5	628.7	125.9	49.8	31.0
Stream04	164.7	413.4	137.7	68.5	458.1	81.8	61.8	30.1
Stream05	155.5	1331.4	108.4	35.5	1503.9	151.5	71.2	28.5
Stream06	58.2	1414.2	81.1	68.4	738.6	52.2	71.8	27.5
Stream07	100.5	1472.6	123.8	104.6	1040.6	74.8	66.9	30.1
Stream08	238.6	430.1	326.0	49.2	397.9	90.8	68.4	27.9
Stream09	151.3	332.4	130.1	123.1	349.2	52.5	59.2	22.4
Min Qi	25.9	127.1	24.4	15.3	106.9	14.2	38.4	22.4
Max Qi	426.2	1513.8	326.0	176.9	1598.0	151.5	71.8	33.0
Avg Qi	162.9	955.7	131.9	75.3	705.4	75.4	59.0	28.9

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 parameter settings.

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 BladeSystem c-Class ProLiant BL460c IB Cluster 16P DC is depicted in Figure 1.1 consists of:

System Components	**System Total**	**Per Node**
HP BladeSystem c7000 Enclosure:	1	n.a.
HP ProLiant BL460c (Database Server):	8	n.a
Processors (Dual-Core Intel Xeon X5160, 3.0 GHz, 1333 MHz FSB)/Cores/Threads:	16/32/32	2/4/4
Memory:	128GB	16GB
HP 36GB 10K SAS 2.5 Hot Plug HDD (OS)	16	2
On board Dual-Port NIC*:	8	1
HP GbE2 Switch:	1	n.a.
HP 4X DDR InfiniBand Mezzanine HCA**:	16	2
HP 4X DDR InfiniBand Switch (Oracle Cluster Interconnect):	2	n.a
Storage Components		
HP BladeSystem c7000 Enclosure:	2	
HP 4X DDR InfiniBand Switch (Storage Interconnect):	2	
HP ProLiant BL465c (1 x Dual-Core AMD Opteron 2214 HE. 2.2 GHz, 2GB Main Memory, 2x HP 36GB 15k 2.5 SAS drive, 1 x HP 4X DDR InfiniBand Mezzanine HCA (Storage Server)***	16	
HP StorageWorks SB40c with 6 x HP 36GB 15k 2.5 SAS drive (Storage Blade):	16	
Total Storage:	5184 GB	

*One Gb NIC port is configured for user connectivity and other for Oracle cluster manager.

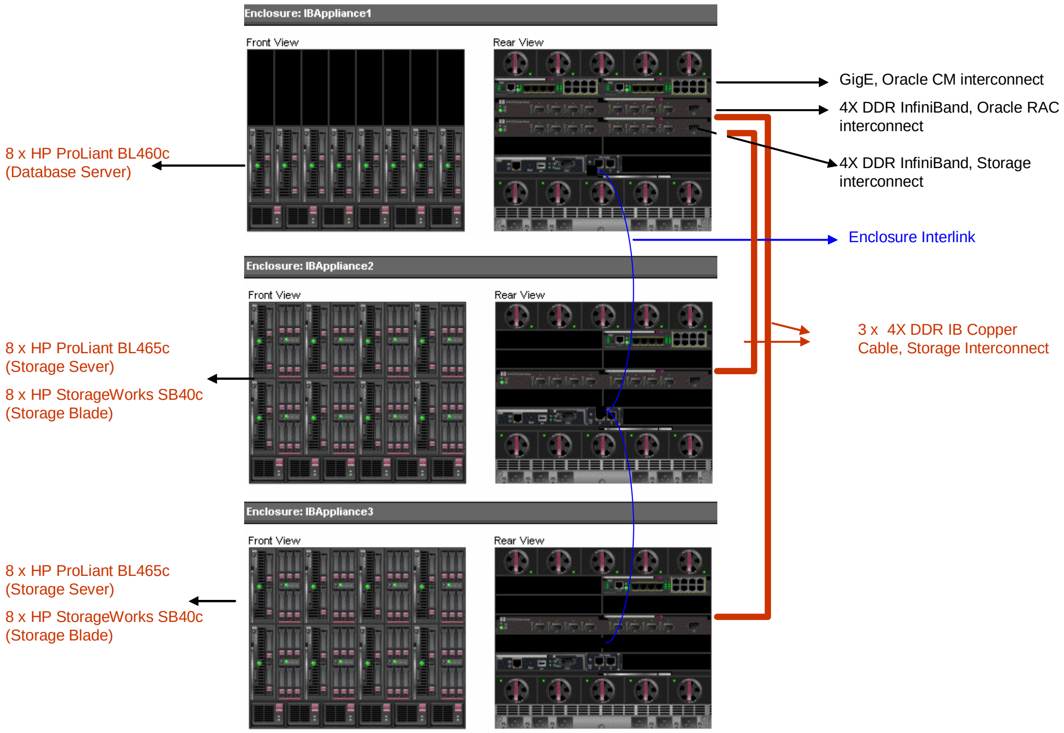
**One HP 4X DDR InfiniBand interface is configured for Oracle cluster interconnect and other for storage interconnect.

***Each Storage Server is configured with (i) one RAID10 array of two 36GB 15k SAS drives (on internal Smart Array Controller E200) and has two logical volumes, one configured for operating system and other configured for database (ii) one Storage Blade with three RAID10 volume of two 36GB 15k SAS drives (on internal Smart Array Controller P400) each has one logical volume, configured for database. The logical volumes are partitioned using Linux which hosted database tables, indexes, redo log files. There are 16 redo log file groups; each redo file group has two members residing on two separate Storage Servers to guarantee no single point of controller/cache failure. The Array accelerator cache was disabled for Smart Array Controller E200 and set to 100% write for Smart Array Controller P400.

The logical volumes on the storage blades are presented to the database servers over HP 4X DDR InfiniBand interface using SRP technology.

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

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 2.6.1 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 2.6.1 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

Atomicity, Consistency and Isolation tests were used from TPC-H publication on HP BladeSystem ProLiant BL480c Cluster 16P DC http://www.tpc.org/tpch/results/tpch_result_detail.asp?id=106121901 submitted on 12/18/06.

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 10 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.

Qualification database was brought up on two nodes. Started test transactions. During the test one of the disks (hosting redo log files) and one of the disks (hosting tables and auxiliary structures) were removed. As the database redo log files, tables and auxiliary data structures were stored on RAID1 volumes and each redo file group has two members residing on two separate storage servers (to guarantee no single point of controller/cache failure), the test continued uninterrupted. Consistency conditions were verified.

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. Qualification database was brought up on two nodes. Started test transactions. During the test power to the HP BladeSystem enclosures were turned off. The power was restored. Started Oracle instance, which automatically recovered the database. 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 as explained in section 4.4.2.

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 Name	Row Count
Region	5
Nation	25
Supplier	3,000,000
Customer	45,000,000
Part	60,000,000
Partsupp	240,000,000
Orders	450,000,000
Lineitem	1,799,989,091

Table 5. 1: Initial Number of Rows

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 BladeSystem c-Class ProLiant BL460c IB Cluster 16P DC is depicted in Figure 1.1 consists of:

System Components	**System Total**	**Per Node**
HP BladeSystem c7000 Enclosure:	1	n.a.
HP ProLiant BL460c (Database Server):	8	n.a
Processors (Dual-Core Intel Xeon X5160, 3.0 GHz, 1333 MHz FSB)/Cores/Threads:	16/32/32	2/4/4
Memory:	128GB	16GB
HP 36GB 10K SAS 2.5 Hot Plug HDD (OS)	16	2
On board Dual-Port NIC*:	8	1
HP GbE2 Switch:	1	n.a.
HP 4X DDR InfiniBand Mezzanine HCA**:	16	2
HP 4X DDR InfiniBand Switch (Oracle Cluster Interconnect):	2	n.a
Storage Components		
HP BladeSystem c7000 Enclosure:	2	
HP 4X DDR InfiniBand Switch (Storage Interconnect):	2	
HP ProLiant BL465c (1 x Dual-Core AMD Opteron 2214 HE. 2.2 GHz, 2GB Main Memory, 2x HP 36GB 15k 2.5 SAS drive, 1 x HP 4X DDR InfiniBand Mezzanine HCA (Storage Server)***	16	
HP StorageWorks SB40c with 6 x HP 36GB 15k 2.5 SAS drive (Storage Blade):	16	
Total Storage:	5184 GB	

*One Gb NIC port is configured for user connectivity and other for Oracle cluster manager.

**One HP 4X DDR InfiniBand interface is configured for Oracle cluster interconnect and other for storage interconnect.

***Each Storage Server is configured with (i) one RAID10 array of two 36GB 15k SAS drives (on internal Smart Array Controller E200) and has two logical volumes, one configured for operating system and other configured for database (ii) one Storage Blade with three RAID10 volume of two 36GB 15k SAS drives (on internal Smart Array Controller P400) each has one logical volume, configured for database. The logical volumes are partitioned using

Linux which hosted database tables, indexes, redo log files. There are 16 redo log file groups; each redo file group has two members residing on two separate Storage Servers to guarantee no single point of controller/cache failure. The Array accelerator cache was disabled for Smart Array Controller E200 and set to 100% write for Smart Array Controller P400.

The logical volumes on the storage blades are presented to the database servers over HP 4X DDR InfiniBand interface using SRP technology.

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

Storage Server	Disk Drives, RAID, Capacity, Controller/Enclosure	Partition	Description	Size
[1..16]	2 x 36GB, RAID1, 26GB, SA E200/Internal * 2 x 36GB, RAID1, 36GB, SA P400/SB40c 2 x 36GB, RAID1, 36GB, SA P400/SB40c 2 x 36GB, RAID1, 36GB, SA P400/SB40c	1	temp	7128MB
		2	tables and indexes	7128MB
		3	redo log	1024MB
		5	undo	1024MB
		6	flat files	6144MB
		7	misc**	640MB

* A 10GB volume on this array was used for OS

**misc -ocr1, quorum 1, control1, control2, sys, sysaux, sp_0, default

Table 5.2: SAN configuration and Database Layout

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 describe 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 1 was used for the entire database and redo log files. Oracle redo file group has two members residing on two separate Storage Servers to guarantee no single point of controller/cache failure.

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 2.6.1 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 1 hour 29 minutes 24 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.

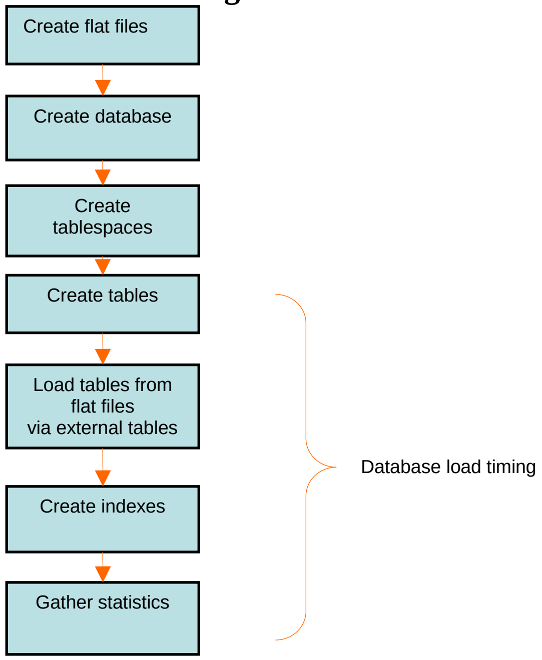
Disk Type	Number of Disks	Total Disk Space	Data Storage Ratio
36GB15k sff SAS Drive 36GB10k sff SAS Drive	128 16	5148GB	17.28

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, and 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.

5.10 Dataset Verification

Verify that the rows in the loaded database after the performance test are correct by comparing some small number
of rows extracted at random from any two files of the corresponding Base, Insert and Delete reference data set files
for each table and the corresponding rows of the database.

Verified according to the specification.

5.11 Referental Intergrity

Verify referential integrity in the database after the initial load.

Verified according to the specification.

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. RF1 Refresh Transaction
2. Stream 00 Execution
3. 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.

Nine 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@300GB	QthH@300GB	QphH@300GB
Run 1	49,541.3	31,675.3	39,613.6
Run 2	51,428.6	30,653.7	39,704.9

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.

No activites performed between Run 1 and Run 2.

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 used in this benchmark.

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. This purchase qualifies for 16% large purchase discount from Hewlett Packard Company. Oracle Database software qualify for an Oracle mandatory E-Business discount.

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.

Availability date is September 1, 2007.

All hardware and software components except the SRP target driver are available now.

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.
PO Box 984 Klamath, CA 95548
Phone: (707) 482-0523
fax: (707) 482-0575

email: LornaL@PerfMetrics.com

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

August 8, 2007

Mr. Raghunath Othayoth Nambiar
Hewlett-Packard Company
20555 SH 249
Houston, TX 77077

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

Platform:	ProLiant BL460c 8 node cluster
Database Manager:	Oracle Database 10g Enterprise Edition R2
Operating System:	Red Hat Enterprise Linux 4

CPU's	Memory	Total Disks	Qpph@ 300GB	QthH@300GB	QphH@300GB
16 Intel Xeon @ 3.00 GHz dual-core	16 GB each node	16 @ 36 GB (OS) 128 @ 36GB	**49,541.3**	**31,675.3**	**39,613.6**

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 300GB using DBGEN. The version of DBGEN was 2.6.0. The references data was verified.
- The qualification database layout was identical to the tested database except for the number and size of the files and nodes.
- The query text was verified to use only compliant variants and minor modifications.
- The executable query text was generated by QGEN and submitted through Oracle's standard interactive interface. The version of QGEN was 2.6.0.

- 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 successfully demonstrated and verified.
- The system pricing was checked for major components and maintenance.
- The executive summary pages of the FDR were verified for accuracy.

Auditor’s Notes:
None.

Sincerely,

draft

Lorna Livingtree
Auditor

Appendix A: Parameter Settings

```
----------------------------
2shut
----------------------------
#!/bin/ksh

. $FRAME_PATH/env
(( dop = 10 ))
(( c=10 ))
echo $START_NODES
for i in $START_NODES
do
(( c=c+1 ))
echo starting instance on node $i
ssh $i -n /home/oracle/frame/bin/tshut abort &
if (( c>=dop )) then
  #wait
  (( c=0 ))
fi
done
wait
----------------------------
2start
----------------------------
#!/bin/ksh

. $FRAME_PATH/env
(( dop = 10 ))
(( c=10 ))
echo $START_NODES
for i in $START_NODES
do
(( c=c+1 ))
echo starting instance on node $i
ssh $i -n /home/oracle/frame/bin/tstart &
if (( c>=dop )) then
  #wait
  (( c=0 ))
fi
done
wait
----------------------------
ib-init.sh
----------------------------
modprobe  ib_srp srp_sg_tablesize=32

# sb1
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8ceb14,ioc_guid=0018feffff8ceb14,dgid
=fe800000000000000018feffff8ceb15,pkey=ffff,service_id=0018feffff8ceb14 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb2
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8cfba0,ioc_guid=0018feffff8cfba0,dgid
=fe800000000000000018feffff8cfba1,pkey=ffff,service_id=0018feffff8cfba0 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb3
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8ceb3c,ioc_guid=0018feffff8ceb3c,dgid
=fe800000000000000018feffff8ceb3d,pkey=ffff,service_id=0018feffff8ceb3c >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb4
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8cfb90,ioc_guid=0018feffff8cfb90,dgid
=fe800000000000000018feffff8cfb91,pkey=ffff,service_id=0018feffff8cfb90 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb5
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8cfbb0,ioc_guid=0018feffff8cfbb0,dgid
=fe800000000000000018feffff8cfbb1,pkey=ffff,service_id=0018feffff8cfbb0  >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb6
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8cfbcc,ioc_guid=0018feffff8cfbcc,dgid=
fe800000000000000018feffff8cfbcd,pkey=ffff,service_id=0018feffff8cfbcc >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb7
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8cdbd8,ioc_guid=0018feffff8cdbd8,dgid
=fe800000000000000018feffff8cdbd9,pkey=ffff,service_id=0018feffff8cdbd8  >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb8
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8c0c14,ioc_guid=0018feffff8c0c14,dgid
=fe800000000000000018feffff8c0c15,pkey=ffff,service_id=0018feffff8c0c14  >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb9
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8cfb94,ioc_guid=0018feffff8cfb94,dgid
=fe800000000000000018feffff8cfb95,pkey=ffff,service_id=0018feffff8cfb94 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb10
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8c0c34,ioc_guid=0018feffff8c0c34,dgid
=fe800000000000000018feffff8c0c35,pkey=ffff,service_id=0018feffff8c0c34 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb11
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8ceb40,ioc_guid=0018feffff8ceb40,dgid
=fe800000000000000018feffff8ceb41,pkey=ffff,service_id=0018feffff8ceb40 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb12
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8c0c60,ioc_guid=0018feffff8c0c60,dgid
=fe800000000000000018feffff8c0c61,pkey=ffff,service_id=0018feffff8c0c60  >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb13
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8ceb78,ioc_guid=0018feffff8ceb78,dgid
=fe800000000000000018feffff8ceb79,pkey=ffff,service_id=0018feffff8ceb78 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb14
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8c0c48,ioc_guid=0018feffff8c0c48,dgid
=fe800000000000000018feffff8c0c49,pkey=ffff,service_id=0018feffff8c0c48 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb15
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8c1c10,ioc_guid=0018feffff8c1c10,dgid
=fe800000000000000018feffff8c1c11,pkey=ffff,service_id=0018feffff8c1c10 >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

# sb16
echo
max_cmd_per_lun=16,max_sect=512,id_ext=0018feffff8ceb5c,ioc_guid=0018feffff8ceb5c,dgid
=fe800000000000000018feffff8ceb5d,pkey=ffff,service_id=0018feffff8ceb5c >
/sys/class/infiniband_srp/srp-mthca1-1/add_target

----------------------------
ib-targ.sh
----------------------------
for j in 0
  do
  for i in 0
    do
    echo 0 > /sys/block/cciss\!c${j}d${i}/queue/iosched/front_merges
    echo 300 > /sys/block/cciss\!c${j}d${i}/queue/iosched/read_expire
    echo 1500 > /sys/block/cciss\!c${j}d${i}/queue/iosched/write_expire
  done
done

for j in 1
  do
  for i in 0 1 2
    do
    echo 0 > /sys/block/cciss\!c${j}d${i}/queue/iosched/front_merges
    echo 300 > /sys/block/cciss\!c${j}d${i}/queue/iosched/read_expire
    echo 1500 > /sys/block/cciss\!c${j}d${i}/queue/iosched/write_expire
  done
done

modprobe scsi_tgt scst_threads=1
modprobe scst_vdisk dedicated_thread=0

echo "open disk0  /dev/cciss/c0d1 BLOCKIO" > /proc/scsi_tgt/vdisk/vdisk
echo "open disk1  /dev/cciss/c1d0 BLOCKIO" > /proc/scsi_tgt/vdisk/vdisk
echo "open disk2  /dev/cciss/c1d1 BLOCKIO" > /proc/scsi_tgt/vdisk/vdisk
echo "open disk3  /dev/cciss/c1d2 BLOCKIO" > /proc/scsi_tgt/vdisk/vdisk
echo "add disk0 0" > /proc/scsi_tgt/groups/Default/devices
echo "add disk1 1" > /proc/scsi_tgt/groups/Default/devices
echo "add disk2 2" > /proc/scsi_tgt/groups/Default/devices
echo "add disk3 3" > /proc/scsi_tgt/groups/Default/devices

modprobe ib_srpt thread=1
----------------------------
init_build.ora
----------------------------
aq_tm_processes                = 0
audit_trail                    = false
compatible                     = 10.2.0.2
control_files                  = (/home/oracle/dev/block/control1)
db_block_checksum              = false
db_block_size                  = 16384
db_cache_size                  = 3500m
db_file_multiblock_read_count  = 64
db_files                       = 500
db_name                        = 10i
db_writer_processes            = 4
dml_locks                      = 5000
global_names                   = false
instance_name                  = raca
log_buffer                     = 4194304
log_checkpoint_timeout                        = 1200
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      = 10.2.0.2.1
parallel_adaptive_multi_user   = true
parallel_execution_message_size = 16384
parallel_max_servers           = 256
parallel_min_servers           = 256
pga_aggregate_target           = 5500m
processes                      = 1000
recovery_parallelism           = 8
replication_dependency_tracking = false
statistics_level               = basic
undo_retention                 = 400000
optimizer_index_cost_adj       = 1400
cpu_count                      = 4
shared_pool_size               = 3000m
filesystemio_options = asynch
undo_management = auto

----------------------------
init_db1.ora
----------------------------
instance_number = 1
thread = 1
undo_management = auto
undo_tablespace = ts_undo1
cluster_database = true
#cluster_interconnects = 10.10.210.1
#cluster_interconnects = 1.10.210.1
cluster_interconnects = 21.15.15.1
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 1
----------------------------
init_db2.ora
----------------------------
instance_number = 2
thread = 2
undo_management = auto
undo_tablespace = ts_undo2
cluster_database = true
#cluster_interconnects = 10.10.210.2
#cluster_interconnects = 1.10.210.2
cluster_interconnects = 21.15.15.2
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 2
----------------------------
init_db3.ora
----------------------------
instance_number = 3
thread = 3
undo_management = auto
undo_tablespace = ts_undo3
cluster_database = true
#cluster_interconnects = 10.10.210.3
#cluster_interconnects = 1.10.210.3
cluster_interconnects = 21.15.15.3
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 3
----------------------------
init_db4.ora
----------------------------
instance_number = 4
thread = 4
undo_management = auto
undo_tablespace = ts_undo4
cluster_database = true
#cluster_interconnects = 10.10.210.4
#cluster_interconnects = 1.10.210.4
cluster_interconnects = 21.15.15.4
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 4
----------------------------
init_db5.ora
----------------------------
instance_number = 5
thread = 5
undo_management = auto
undo_tablespace = ts_undo5
cluster_database = true
#cluster_interconnects = 10.10.210.5
#cluster_interconnects = 1.10.210.5
cluster_interconnects = 21.15.15.5
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 5
----------------------------
init_db6.ora
----------------------------
instance_number = 6
thread = 6
undo_management = auto
undo_tablespace = ts_undo6
cluster_database = true
#cluster_interconnects = 10.10.210.6
#cluster_interconnects = 1.10.210.6
cluster_interconnects = 21.15.15.6
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 6
----------------------------
init_db7.ora
----------------------------
instance_number = 7
thread = 7
undo_management = auto
undo_tablespace = ts_undo7
cluster_database = true
#cluster_interconnects = 10.10.210.7
#cluster_interconnects = 1.10.210.7
cluster_interconnects = 21.15.15.7
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 7
----------------------------
init_db8.ora
----------------------------
instance_number = 8
thread = 8
undo_management = auto
undo_tablespace = ts_undo8
cluster_database = true
#cluster_interconnects = 10.10.210.8
#cluster_interconnects = 1.10.210.8
cluster_interconnects = 21.15.15.8
ifile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora
#cluster_database_instances = 8
----------------------------
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
mknod /home/oracle/dev/raw/ocr1 c 162 1
mknod /home/oracle/dev/raw/quorum1 c 162 2
raw /home/oracle/dev/raw/ocr1 /dev/sdd7
raw /home/oracle/dev/raw/quorum1 /dev/sdh7

mknod /home/oracle/dev/block/t_1 c 162 3
mknod /home/oracle/dev/block/lo_1 c 162 4
mknod /home/oracle/dev/block/log_1 c 162 5
raw /home/oracle/dev/block/t_1 /dev/sda1
raw /home/oracle/dev/block/lo_1 /dev/sda2
raw /home/oracle/dev/block/log_1 /dev/sda3
mknod /home/oracle/dev/block/t_2 c 162 6
mknod /home/oracle/dev/block/lo_2 c 162 7
mknod /home/oracle/dev/block/log_2 c 162 8
raw /home/oracle/dev/block/t_2 /dev/sdb1
raw /home/oracle/dev/block/lo_2 /dev/sdb2
raw /home/oracle/dev/block/log_2 /dev/sdb3
mknod /home/oracle/dev/block/t_3 c 162 9
mknod /home/oracle/dev/block/lo_3 c 162 10
mknod /home/oracle/dev/block/log_3 c 162 11
raw /home/oracle/dev/block/t_3 /dev/sdc1
raw /home/oracle/dev/block/lo_3 /dev/sdc2
raw /home/oracle/dev/block/log_3 /dev/sdc3
mknod /home/oracle/dev/block/t_4 c 162 12
mknod /home/oracle/dev/block/lo_4 c 162 13
mknod /home/oracle/dev/block/log_4 c 162 14
raw /home/oracle/dev/block/t_4 /dev/sdd1
raw /home/oracle/dev/block/lo_4 /dev/sdd2
raw /home/oracle/dev/block/log_4 /dev/sdd3
mknod /home/oracle/dev/block/t_5 c 162 15
mknod /home/oracle/dev/block/lo_5 c 162 16
mknod /home/oracle/dev/block/log_5 c 162 17
raw /home/oracle/dev/block/t_5 /dev/sde1
raw /home/oracle/dev/block/lo_5 /dev/sde2
raw /home/oracle/dev/block/log_5 /dev/sde3
mknod /home/oracle/dev/block/t_6 c 162 18
mknod /home/oracle/dev/block/lo_6 c 162 19
mknod /home/oracle/dev/block/log_6 c 162 20
raw /home/oracle/dev/block/t_6 /dev/sdf1
raw /home/oracle/dev/block/lo_6 /dev/sdf2
raw /home/oracle/dev/block/log_6 /dev/sdf3
mknod /home/oracle/dev/block/t_7 c 162 21
mknod /home/oracle/dev/block/lo_7 c 162 22
mknod /home/oracle/dev/block/log_7 c 162 23
raw /home/oracle/dev/block/t_7 /dev/sdg1
raw /home/oracle/dev/block/lo_7 /dev/sdg2
raw /home/oracle/dev/block/log_7 /dev/sdg3
mknod /home/oracle/dev/block/t_8 c 162 24
mknod /home/oracle/dev/block/lo_8 c 162 25
mknod /home/oracle/dev/block/log_8 c 162 26
raw /home/oracle/dev/block/t_8 /dev/sdh1
raw /home/oracle/dev/block/lo_8 /dev/sdh2
raw /home/oracle/dev/block/log_8 /dev/sdh3
mknod /home/oracle/dev/block/t_9 c 162 27
mknod /home/oracle/dev/block/lo_9 c 162 28
mknod /home/oracle/dev/block/log_9 c 162 29
raw /home/oracle/dev/block/t_9 /dev/sdi1
raw /home/oracle/dev/block/lo_9 /dev/sdi2
raw /home/oracle/dev/block/log_9 /dev/sdi3
mknod /home/oracle/dev/block/t_10 c 162 30
mknod /home/oracle/dev/block/lo_10 c 162 31
mknod /home/oracle/dev/block/log_10 c 162 32
raw /home/oracle/dev/block/t_10 /dev/sdj1
raw /home/oracle/dev/block/lo_10 /dev/sdj2
raw /home/oracle/dev/block/log_10 /dev/sdj3
mknod /home/oracle/dev/block/t_11 c 162 33
mknod /home/oracle/dev/block/lo_11 c 162 34
mknod /home/oracle/dev/block/log_11 c 162 35
raw /home/oracle/dev/block/t_11 /dev/sdk1
raw /home/oracle/dev/block/lo_11 /dev/sdk2
raw /home/oracle/dev/block/log_11 /dev/sdk3
mknod /home/oracle/dev/block/t_12 c 162 36
mknod /home/oracle/dev/block/lo_12 c 162 37
mknod /home/oracle/dev/block/log_12 c 162 38
raw /home/oracle/dev/block/t_12 /dev/sdl1
raw /home/oracle/dev/block/lo_12 /dev/sdl2
raw /home/oracle/dev/block/log_12 /dev/sdl3
mknod /home/oracle/dev/block/t_13 c 162 39
mknod /home/oracle/dev/block/lo_13 c 162 40
mknod /home/oracle/dev/block/log_13 c 162 41
raw /home/oracle/dev/block/t_13 /dev/sdm1
raw /home/oracle/dev/block/lo_13 /dev/sdm2
raw /home/oracle/dev/block/log_13 /dev/sdm3
mknod /home/oracle/dev/block/t_14 c 162 42
mknod /home/oracle/dev/block/lo_14 c 162 43
mknod /home/oracle/dev/block/log_14 c 162 44
raw /home/oracle/dev/block/t_14 /dev/sdn1
raw /home/oracle/dev/block/lo_14 /dev/sdn2
raw /home/oracle/dev/block/log_14 /dev/sdn3
mknod /home/oracle/dev/block/t_15 c 162 45
mknod /home/oracle/dev/block/lo_15 c 162 46
mknod /home/oracle/dev/block/log_15 c 162 47
raw /home/oracle/dev/block/t_15 /dev/sdo1
raw /home/oracle/dev/block/lo_15 /dev/sdo2
raw /home/oracle/dev/block/log_15 /dev/sdo3
mknod /home/oracle/dev/block/t_16 c 162 48
mknod /home/oracle/dev/block/lo_16 c 162 49
mknod /home/oracle/dev/block/log_16 c 162 50
raw /home/oracle/dev/block/t_16 /dev/sdp1
raw /home/oracle/dev/block/lo_16 /dev/sdp2
raw /home/oracle/dev/block/log_16 /dev/sdp3
mknod /home/oracle/dev/block/t_17 c 162 51
mknod /home/oracle/dev/block/lo_17 c 162 52
mknod /home/oracle/dev/block/log_17 c 162 53
raw /home/oracle/dev/block/t_17 /dev/sdq1
raw /home/oracle/dev/block/lo_17 /dev/sdq2
raw /home/oracle/dev/block/log_17 /dev/sdq3
mknod /home/oracle/dev/block/t_18 c 162 54
mknod /home/oracle/dev/block/lo_18 c 162 55
mknod /home/oracle/dev/block/log_18 c 162 56
raw /home/oracle/dev/block/t_18 /dev/sdr1
raw /home/oracle/dev/block/lo_18 /dev/sdr2
raw /home/oracle/dev/block/log_18 /dev/sdr3
mknod /home/oracle/dev/block/t_19 c 162 57
mknod /home/oracle/dev/block/lo_19 c 162 58
mknod /home/oracle/dev/block/log_19 c 162 59
raw /home/oracle/dev/block/t_19 /dev/sds1
raw /home/oracle/dev/block/lo_19 /dev/sds2
raw /home/oracle/dev/block/log_19 /dev/sds3
mknod /home/oracle/dev/block/t_20 c 162 60
mknod /home/oracle/dev/block/lo_20 c 162 61
mknod /home/oracle/dev/block/log_20 c 162 62
raw /home/oracle/dev/block/t_20 /dev/sdt1
raw /home/oracle/dev/block/lo_20 /dev/sdt2
raw /home/oracle/dev/block/log_20 /dev/sdt3
mknod /home/oracle/dev/block/t_21 c 162 63
mknod /home/oracle/dev/block/lo_21 c 162 64
mknod /home/oracle/dev/block/log_21 c 162 65
raw /home/oracle/dev/block/t_21 /dev/sdu1
raw /home/oracle/dev/block/lo_21 /dev/sdu2
raw /home/oracle/dev/block/log_21 /dev/sdu3
mknod /home/oracle/dev/block/t_22 c 162 66
mknod /home/oracle/dev/block/lo_22 c 162 67
mknod /home/oracle/dev/block/log_22 c 162 68
raw /home/oracle/dev/block/t_22 /dev/sdv1
raw /home/oracle/dev/block/lo_22 /dev/sdv2
raw /home/oracle/dev/block/log_22 /dev/sdv3
mknod /home/oracle/dev/block/t_23 c 162 69
mknod /home/oracle/dev/block/lo_23 c 162 70
mknod /home/oracle/dev/block/log_23 c 162 71
raw /home/oracle/dev/block/t_23 /dev/sdw1
raw /home/oracle/dev/block/lo_23 /dev/sdw2
raw /home/oracle/dev/block/log_23 /dev/sdw3
mknod /home/oracle/dev/block/t_24 c 162 72
mknod /home/oracle/dev/block/lo_24 c 162 73
mknod /home/oracle/dev/block/log_24 c 162 74
raw /home/oracle/dev/block/t_24 /dev/sdx1
raw /home/oracle/dev/block/lo_24 /dev/sdx2
raw /home/oracle/dev/block/log_24 /dev/sdx3
mknod /home/oracle/dev/block/t_25 c 162 75
mknod /home/oracle/dev/block/lo_25 c 162 76
mknod /home/oracle/dev/block/log_25 c 162 77
raw /home/oracle/dev/block/t_25 /dev/sdy1
raw /home/oracle/dev/block/lo_25 /dev/sdy2
raw /home/oracle/dev/block/log_25 /dev/sdy3
mknod /home/oracle/dev/block/t_26 c 162 78
mknod /home/oracle/dev/block/lo_26 c 162 79
mknod /home/oracle/dev/block/log_26 c 162 80
raw /home/oracle/dev/block/t_26 /dev/sdz1
raw /home/oracle/dev/block/lo_26 /dev/sdz2
raw /home/oracle/dev/block/log_26 /dev/sdz3
mknod /home/oracle/dev/block/t_27 c 162 81
mknod /home/oracle/dev/block/lo_27 c 162 82
mknod /home/oracle/dev/block/log_27 c 162 83
raw /home/oracle/dev/block/t_27 /dev/sdaa1
raw /home/oracle/dev/block/lo_27 /dev/sdaa2
raw /home/oracle/dev/block/log_27 /dev/sdaa3
mknod /home/oracle/dev/block/t_28 c 162 84
mknod /home/oracle/dev/block/lo_28 c 162 85
mknod /home/oracle/dev/block/log_28 c 162 86
raw /home/oracle/dev/block/t_28 /dev/sdab1
raw /home/oracle/dev/block/lo_28 /dev/sdab2
raw /home/oracle/dev/block/log_28 /dev/sdab3
mknod /home/oracle/dev/block/t_29 c 162 87
mknod /home/oracle/dev/block/lo_29 c 162 88
mknod /home/oracle/dev/block/log_29 c 162 89
raw /home/oracle/dev/block/t_29 /dev/sdac1
raw /home/oracle/dev/block/lo_29 /dev/sdac2
raw /home/oracle/dev/block/log_29 /dev/sdac3
mknod /home/oracle/dev/block/t_30 c 162 90
mknod /home/oracle/dev/block/lo_30 c 162 91
mknod /home/oracle/dev/block/log_30 c 162 92
raw /home/oracle/dev/block/t_30 /dev/sdad1
raw /home/oracle/dev/block/lo_30 /dev/sdad2
raw /home/oracle/dev/block/log_30 /dev/sdad3
mknod /home/oracle/dev/block/t_31 c 162 93
mknod /home/oracle/dev/block/lo_31 c 162 94
mknod /home/oracle/dev/block/log_31 c 162 95
raw /home/oracle/dev/block/t_31 /dev/sdae1
raw /home/oracle/dev/block/lo_31 /dev/sdae2
raw /home/oracle/dev/block/log_31 /dev/sdae3
mknod /home/oracle/dev/block/t_32 c 162 96
mknod /home/oracle/dev/block/lo_32 c 162 97
mknod /home/oracle/dev/block/log_32 c 162 98
raw /home/oracle/dev/block/t_32 /dev/sdaf1
raw /home/oracle/dev/block/lo_32 /dev/sdaf2
raw /home/oracle/dev/block/log_32 /dev/sdaf3
mknod /home/oracle/dev/block/t_33 c 162 99
mknod /home/oracle/dev/block/lo_33 c 162 100
mknod /home/oracle/dev/block/log_33 c 162 101
raw /home/oracle/dev/block/t_33 /dev/sdag1
raw /home/oracle/dev/block/lo_33 /dev/sdag2
raw /home/oracle/dev/block/log_33 /dev/sdag3
mknod /home/oracle/dev/block/t_34 c 162 102
mknod /home/oracle/dev/block/lo_34 c 162 103
mknod /home/oracle/dev/block/log_34 c 162 104
raw /home/oracle/dev/block/t_34 /dev/sdah1
raw /home/oracle/dev/block/lo_34 /dev/sdah2
raw /home/oracle/dev/block/log_34 /dev/sdah3
mknod /home/oracle/dev/block/t_35 c 162 105
mknod /home/oracle/dev/block/lo_35 c 162 106
mknod /home/oracle/dev/block/log_35 c 162 107
raw /home/oracle/dev/block/t_35 /dev/sdai1
raw /home/oracle/dev/block/lo_35 /dev/sdai2
raw /home/oracle/dev/block/log_35 /dev/sdai3
mknod /home/oracle/dev/block/t_36 c 162 108
mknod /home/oracle/dev/block/lo_36 c 162 109
mknod /home/oracle/dev/block/log_36 c 162 110
raw /home/oracle/dev/block/t_36 /dev/sdaj1
raw /home/oracle/dev/block/lo_36 /dev/sdaj2
raw /home/oracle/dev/block/log_36 /dev/sdaj3
mknod /home/oracle/dev/block/t_37 c 162 111
mknod /home/oracle/dev/block/lo_37 c 162 112
mknod /home/oracle/dev/block/log_37 c 162 113
raw /home/oracle/dev/block/t_37 /dev/sdak1
raw /home/oracle/dev/block/lo_37 /dev/sdak2
raw /home/oracle/dev/block/log_37 /dev/sdak3
mknod /home/oracle/dev/block/t_38 c 162 114
mknod /home/oracle/dev/block/lo_38 c 162 115
mknod /home/oracle/dev/block/log_38 c 162 116
raw /home/oracle/dev/block/t_38 /dev/sdal1
raw /home/oracle/dev/block/lo_38 /dev/sdal2
raw /home/oracle/dev/block/log_38 /dev/sdal3
mknod /home/oracle/dev/block/t_39 c 162 117
mknod /home/oracle/dev/block/lo_39 c 162 118
mknod /home/oracle/dev/block/log_39 c 162 119
raw /home/oracle/dev/block/t_39 /dev/sdam1
raw /home/oracle/dev/block/lo_39 /dev/sdam2
raw /home/oracle/dev/block/log_39 /dev/sdam3
mknod /home/oracle/dev/block/t_40 c 162 120
mknod /home/oracle/dev/block/lo_40 c 162 121
mknod /home/oracle/dev/block/log_40 c 162 122
raw /home/oracle/dev/block/t_40 /dev/sdan1
raw /home/oracle/dev/block/lo_40 /dev/sdan2
raw /home/oracle/dev/block/log_40 /dev/sdan3
mknod /home/oracle/dev/block/t_41 c 162 123
mknod /home/oracle/dev/block/lo_41 c 162 124
mknod /home/oracle/dev/block/log_41 c 162 125
raw /home/oracle/dev/block/t_41 /dev/sdao1
raw /home/oracle/dev/block/lo_41 /dev/sdao2
raw /home/oracle/dev/block/log_41 /dev/sdao3
mknod /home/oracle/dev/block/t_42 c 162 126
mknod /home/oracle/dev/block/lo_42 c 162 127
mknod /home/oracle/dev/block/log_42 c 162 128
raw /home/oracle/dev/block/t_42 /dev/sdap1
raw /home/oracle/dev/block/lo_42 /dev/sdap2
raw /home/oracle/dev/block/log_42 /dev/sdap3
mknod /home/oracle/dev/block/t_43 c 162 129
mknod /home/oracle/dev/block/lo_43 c 162 130
mknod /home/oracle/dev/block/log_43 c 162 131
raw /home/oracle/dev/block/t_43 /dev/sdaq1
raw /home/oracle/dev/block/lo_43 /dev/sdaq2
raw /home/oracle/dev/block/log_43 /dev/sdaq3
mknod /home/oracle/dev/block/t_44 c 162 132
mknod /home/oracle/dev/block/lo_44 c 162 133
mknod /home/oracle/dev/block/log_44 c 162 134
raw /home/oracle/dev/block/t_44 /dev/sdar1
raw /home/oracle/dev/block/lo_44 /dev/sdar2
raw /home/oracle/dev/block/log_44 /dev/sdar3
mknod /home/oracle/dev/block/t_45 c 162 135
mknod /home/oracle/dev/block/lo_45 c 162 136
mknod /home/oracle/dev/block/log_45 c 162 137
raw /home/oracle/dev/block/t_45 /dev/sdas1
raw /home/oracle/dev/block/lo_45 /dev/sdas2
raw /home/oracle/dev/block/log_45 /dev/sdas3
mknod /home/oracle/dev/block/t_46 c 162 138
mknod /home/oracle/dev/block/lo_46 c 162 139
mknod /home/oracle/dev/block/log_46 c 162 140
raw /home/oracle/dev/block/t_46 /dev/sdat1
raw /home/oracle/dev/block/lo_46 /dev/sdat2
raw /home/oracle/dev/block/log_46 /dev/sdat3
mknod /home/oracle/dev/block/t_47 c 162 141
mknod /home/oracle/dev/block/lo_47 c 162 142
mknod /home/oracle/dev/block/log_47 c 162 143
raw /home/oracle/dev/block/t_47 /dev/sdau1
raw /home/oracle/dev/block/lo_47 /dev/sdau2
raw /home/oracle/dev/block/log_47 /dev/sdau3
mknod /home/oracle/dev/block/t_48 c 162 144
mknod /home/oracle/dev/block/lo_48 c 162 145
mknod /home/oracle/dev/block/log_48 c 162 146
raw /home/oracle/dev/block/t_48 /dev/sdav1
raw /home/oracle/dev/block/lo_48 /dev/sdav2
raw /home/oracle/dev/block/log_48 /dev/sdav3
mknod /home/oracle/dev/block/t_49 c 162 147
mknod /home/oracle/dev/block/lo_49 c 162 148
mknod /home/oracle/dev/block/log_49 c 162 149
raw /home/oracle/dev/block/t_49 /dev/sdaw1
raw /home/oracle/dev/block/lo_49 /dev/sdaw2
raw /home/oracle/dev/block/log_49 /dev/sdaw3
mknod /home/oracle/dev/block/t_50 c 162 150
mknod /home/oracle/dev/block/lo_50 c 162 151
```

```
mknod /home/oracle/dev/block/log_50 c 162 152
raw /home/oracle/dev/block/t_50 /dev/sdax1
raw /home/oracle/dev/block/lo_50 /dev/sdax2
raw /home/oracle/dev/block/log_50 /dev/sdax3
mknod /home/oracle/dev/block/t_51 c 162 153
mknod /home/oracle/dev/block/lo_51 c 162 154
mknod /home/oracle/dev/block/log_51 c 162 155
raw /home/oracle/dev/block/t_51 /dev/sday1
raw /home/oracle/dev/block/lo_51 /dev/sday2
raw /home/oracle/dev/block/log_51 /dev/sday3
mknod /home/oracle/dev/block/t_52 c 162 156
mknod /home/oracle/dev/block/lo_52 c 162 157
mknod /home/oracle/dev/block/log_52 c 162 158
raw /home/oracle/dev/block/t_52 /dev/sdaz1
raw /home/oracle/dev/block/lo_52 /dev/sdaz2
raw /home/oracle/dev/block/log_52 /dev/sdaz3
mknod /home/oracle/dev/block/t_53 c 162 159
mknod /home/oracle/dev/block/lo_53 c 162 160
mknod /home/oracle/dev/block/log_53 c 162 161
raw /home/oracle/dev/block/t_53 /dev/sdba1
raw /home/oracle/dev/block/lo_53 /dev/sdba2
raw /home/oracle/dev/block/log_53 /dev/sdba3
mknod /home/oracle/dev/block/t_54 c 162 162
mknod /home/oracle/dev/block/lo_54 c 162 163
mknod /home/oracle/dev/block/log_54 c 162 164
raw /home/oracle/dev/block/t_54 /dev/sdbb1
raw /home/oracle/dev/block/lo_54 /dev/sdbb2
raw /home/oracle/dev/block/log_54 /dev/sdbb3
mknod /home/oracle/dev/block/t_55 c 162 165
mknod /home/oracle/dev/block/lo_55 c 162 166
mknod /home/oracle/dev/block/log_55 c 162 167
raw /home/oracle/dev/block/t_55 /dev/sdbc1
raw /home/oracle/dev/block/lo_55 /dev/sdbc2
raw /home/oracle/dev/block/log_55 /dev/sdbc3
mknod /home/oracle/dev/block/t_56 c 162 168
mknod /home/oracle/dev/block/lo_56 c 162 169
mknod /home/oracle/dev/block/log_56 c 162 170
raw /home/oracle/dev/block/t_56 /dev/sdbd1
raw /home/oracle/dev/block/lo_56 /dev/sdbd2
raw /home/oracle/dev/block/log_56 /dev/sdbd3
mknod /home/oracle/dev/block/t_57 c 162 171
mknod /home/oracle/dev/block/lo_57 c 162 172
mknod /home/oracle/dev/block/log_57 c 162 173
raw /home/oracle/dev/block/t_57 /dev/sdbe1
raw /home/oracle/dev/block/lo_57 /dev/sdbe2
raw /home/oracle/dev/block/log_57 /dev/sdbe3
mknod /home/oracle/dev/block/t_58 c 162 174
mknod /home/oracle/dev/block/lo_58 c 162 175
mknod /home/oracle/dev/block/log_58 c 162 176
raw /home/oracle/dev/block/t_58 /dev/sdbf1
raw /home/oracle/dev/block/lo_58 /dev/sdbf2
raw /home/oracle/dev/block/log_58 /dev/sdbf3
mknod /home/oracle/dev/block/t_59 c 162 177
mknod /home/oracle/dev/block/lo_59 c 162 178
mknod /home/oracle/dev/block/log_59 c 162 179
raw /home/oracle/dev/block/t_59 /dev/sdbg1
raw /home/oracle/dev/block/lo_59 /dev/sdbg2
raw /home/oracle/dev/block/log_59 /dev/sdbg3
mknod /home/oracle/dev/block/t_60 c 162 180
mknod /home/oracle/dev/block/lo_60 c 162 181
mknod /home/oracle/dev/block/log_60 c 162 182
raw /home/oracle/dev/block/t_60 /dev/sdbh1
raw /home/oracle/dev/block/lo_60 /dev/sdbh2
raw /home/oracle/dev/block/log_60 /dev/sdbh3
mknod /home/oracle/dev/block/t_61 c 162 183
mknod /home/oracle/dev/block/lo_61 c 162 184
mknod /home/oracle/dev/block/log_61 c 162 185
raw /home/oracle/dev/block/t_61 /dev/sdbi1
raw /home/oracle/dev/block/lo_61 /dev/sdbi2
raw /home/oracle/dev/block/log_61 /dev/sdbi3
mknod /home/oracle/dev/block/t_62 c 162 186
mknod /home/oracle/dev/block/lo_62 c 162 187
mknod /home/oracle/dev/block/log_62 c 162 188
raw /home/oracle/dev/block/t_62 /dev/sdbj1
raw /home/oracle/dev/block/lo_62 /dev/sdbj2
raw /home/oracle/dev/block/log_62 /dev/sdbj3
mknod /home/oracle/dev/block/t_63 c 162 189
mknod /home/oracle/dev/block/lo_63 c 162 190
mknod /home/oracle/dev/block/log_63 c 162 191
raw /home/oracle/dev/block/t_63 /dev/sdbk1
raw /home/oracle/dev/block/lo_63 /dev/sdbk2
raw /home/oracle/dev/block/log_63 /dev/sdbk3
mknod /home/oracle/dev/block/t_64 c 162 192
mknod /home/oracle/dev/block/lo_64 c 162 193
mknod /home/oracle/dev/block/log_64 c 162 194
raw /home/oracle/dev/block/t_64 /dev/sdbl1
raw /home/oracle/dev/block/lo_64 /dev/sdbl2
raw /home/oracle/dev/block/log_64 /dev/sdbl3
mknod /home/oracle/dev/block/sys1 c 162 195
mknod /home/oracle/dev/block/sys2 c 162 196
mknod /home/oracle/dev/block/sysaux1 c 162 197
mknod /home/oracle/dev/block/sp_1 c 162 198
mknod /home/oracle/dev/block/control1 c 162 199
mknod /home/oracle/dev/block/control2 c 162 200
raw /home/oracle/dev/block/sys1 /dev/sda5
raw /home/oracle/dev/block/sys2 /dev/sdb5
raw /home/oracle/dev/block/sysaux1 /dev/sdc5
raw /home/oracle/dev/block/sp_1 /dev/sdd5
raw /home/oracle/dev/block/control1 /dev/sde5
raw /home/oracle/dev/block/control2 /dev/sdf5
mknod /home/oracle/dev/block/undo_1 c 162 201
mknod /home/oracle/dev/block/undo_2 c 162 202
mknod /home/oracle/dev/block/undo_3 c 162 203
mknod /home/oracle/dev/block/undo_4 c 162 204
mknod /home/oracle/dev/block/undo_5 c 162 205
mknod /home/oracle/dev/block/undo_6 c 162 206
mknod /home/oracle/dev/block/undo_7 c 162 207
mknod /home/oracle/dev/block/undo_8 c 162 208
raw /home/oracle/dev/block/undo_1 /dev/sdi5
raw /home/oracle/dev/block/undo_2 /dev/sdh5
raw /home/oracle/dev/block/undo_3 /dev/sdg5
raw /home/oracle/dev/block/undo_4 /dev/sdj5
raw /home/oracle/dev/block/undo_5 /dev/sdk5
raw /home/oracle/dev/block/undo_6 /dev/sdl5
raw /home/oracle/dev/block/undo_7 /dev/sdm5
raw /home/oracle/dev/block/undo_8 /dev/sdn5
mknod /home/oracle/dev/block/default c 162 209
raw /home/oracle/dev/block/default /dev/sdo5

mknod /home/oracle/dev/block/1g_1 c 162 210
mknod /home/oracle/dev/block/1g_2 c 162 211
mknod /home/oracle/dev/block/1g_3 c 162 212
mknod /home/oracle/dev/block/1g_4 c 162 213
mknod /home/oracle/dev/block/1g_5 c 162 214
mknod /home/oracle/dev/block/1g_6 c 162 215
mknod /home/oracle/dev/block/1g_7 c 162 216
mknod /home/oracle/dev/block/1g_8 c 162 217
mknod /home/oracle/dev/block/1g_9 c 162 218
mknod /home/oracle/dev/block/1g_10 c 162 219
mknod /home/oracle/dev/block/1g_11 c 162 220
mknod /home/oracle/dev/block/1g_12 c 162 221
mknod /home/oracle/dev/block/1g_13 c 162 222
mknod /home/oracle/dev/block/1g_14 c 162 223
mknod /home/oracle/dev/block/1g_15 c 162 224
mknod /home/oracle/dev/block/1g_16 c 162 225

raw /home/oracle/dev/block/1g_1 /dev/sdi7
raw /home/oracle/dev/block/1g_2 /dev/sdj7
raw /home/oracle/dev/block/1g_3 /dev/sdk7
raw /home/oracle/dev/block/1g_4 /dev/sdl7
raw /home/oracle/dev/block/1g_5 /dev/sdm7
raw /home/oracle/dev/block/1g_6 /dev/sdn7
raw /home/oracle/dev/block/1g_7 /dev/sdo7
raw /home/oracle/dev/block/1g_8 /dev/sdp7
raw /home/oracle/dev/block/1g_9 /dev/sdq7
raw /home/oracle/dev/block/1g_10 /dev/sdr7
raw /home/oracle/dev/block/1g_11 /dev/sds7
raw /home/oracle/dev/block/1g_12 /dev/sdt7
raw /home/oracle/dev/block/1g_13 /dev/sdu7
raw /home/oracle/dev/block/1g_14 /dev/sdv7
raw /home/oracle/dev/block/1g_15 /dev/sdw7
raw /home/oracle/dev/block/1g_16 /dev/sdx7

chown -R oracle:oracle /home/oracle/dev/block/
----------------------------
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

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# 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

fs.file-max = 4194304

kernel.sem = 250        32000   100      128

fs.aio-max-nr = 4194304
kernel.shmmax = 4563402752
kernel.shmall = 409600000

net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 2097152
net.core.wmem_default = 2097152
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_max_syn_backlog = 1536
net.core.netdev_max_backlog = 3000
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack =1
net.ipv4.tcp_window_scaling = 1
----------------------------
tstart
----------------------------
#!/bin/ksh
#
# $Header: tstart.sh 08-aug-99.18:05:50 mpoess Exp $
#
# tstart.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      tstart.sh
#
#    DESCRIPTION
#      starts a database with a specific init.ora or uses the default.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#
#!/bin/ksh

#set -x

DIR=`pwd`
cd $ORACLE_HOME/dbs

if [ "$1" != "" ]; then
  PFILE="pfile=$ORACLE_HOME/dbs/$1.ora"
else
  PFILE="pfile=$ORACLE_HOME/dbs/init_$ORACLE_SID.ora"
fi


sqlplus /NOLOG << !
connect / as sysdba
rem shutdown immediate
startup $PFILE
--alter system set db_file_multiblock_read_count = 48;
execute dbms_scheduler.disable('AUTO_TASKS_JOB_CLASS');
--alter system set statistics_level = basic;

exit
!

cd $DIR
```

Appendix B: Database Build Scripts

```
----------------------------
addts.sh
----------------------------
sqlplus /NOLOG<<!
connect / as sysdba;
alter tablespace $1 add datafile '$2' size $3 reuse;
!

----------------------------
addtts.sh
----------------------------
sqlplus /NOLOG<<!
connect / as sysdba;
alter tablespace $1 add tempfile '$2' size $3 reuse;
!

----------------------------
addundolog.sh
----------------------------
one=$1
((one=one*2-1))
((two=one+1))
echo start creating undo and log for node $1 `date`
sqlplus /NOLOG <<!
connect / as sysdba;
create undo tablespace ts_undo${1} datafile '/home/oracle/dev/block/undo_${1}' size 1536m
reuse;
alter database add logfile thread ${1} '/home/oracle/dev/block/log_${one}' size 1024m reuse,
                                 '/home/oracle/dev/block/log_${two}' size 1024m reuse,
                                 '/home/oracle/dev/block/log_${one}_a' size 1024m reuse,
                                 '/home/oracle/dev/block/log_${two}_a' size 1024m reuse;
alter database enable public thread ${1};
!
echo end creating undo and log for node $1 `date`
----------------------------
create_dir.sh
----------------------------
sqlplus /NOLOG <<!
connect / as sysdba
connect tpch/tpch;
drop directory ff$1;
create directory ff$1 as '/home/oracle/dev/ff_$1';

----------------------------
create_et2_driver.sh
----------------------------
i=0;

while (( i<32 ));do
  (( i=i+1 ))
  /home/oracle/kit/update/scripts/create_et2.sh ${i}
done
----------------------------
create_et2.sh
----------------------------
. $KIT_DIR/env

PAR_HINT=32;
PAR_HINT=64;
SETNUM=$1
sqlplus /NOLOG << !
connect tpch/tpch;
set timing on
set serveroutput on
set echo on
drop directory data_dir;
drop directory ff1;
drop directory ff2;
drop directory ff3;
drop directory ff4;
drop directory ff5;
drop directory ff6;
drop directory ff7;
drop directory ff8;
drop directory ff9;
drop directory ff10;
drop directory ff11;
drop directory ff12;
drop directory ff13;
drop directory ff14;
drop directory ff15;
drop directory ff16;
drop directory ff17;
drop directory ff18;
drop directory ff19;
drop directory ff20;
drop directory ff21;
drop directory ff22;
drop directory ff23;
drop directory ff24;
drop directory ff25;
drop directory ff26;
drop directory ff27;
drop directory ff28;
drop directory ff29;
drop directory ff30;
drop directory ff31;
drop directory ff32;

create directory  data_dir as  '/home/oracle/dev/ff_1/update_sets';

create directory ff1 as  '/home/oracle/dev/ff_1/update_sets';
create directory ff2 as  '/home/oracle/dev/ff_2/update_sets';
create directory ff3 as  '/home/oracle/dev/ff_3/update_sets';
create directory ff4 as  '/home/oracle/dev/ff_4/update_sets';
create directory ff5 as  '/home/oracle/dev/ff_5/update_sets';
create directory ff6 as  '/home/oracle/dev/ff_6/update_sets';
create directory ff7 as  '/home/oracle/dev/ff_7/update_sets';
create directory ff8 as  '/home/oracle/dev/ff_8/update_sets';
create directory ff9 as  '/home/oracle/dev/ff_9/update_sets';
create directory ff10 as  '/home/oracle/dev/ff_10/update_sets';
create directory ff11 as  '/home/oracle/dev/ff_11/update_sets';
create directory ff12 as  '/home/oracle/dev/ff_12/update_sets';
create directory ff13 as  '/home/oracle/dev/ff_13/update_sets';
create directory ff14 as  '/home/oracle/dev/ff_14/update_sets';
create directory ff15 as  '/home/oracle/dev/ff_15/update_sets';
create directory ff16 as  '/home/oracle/dev/ff_16/update_sets';
create directory ff17 as  '/home/oracle/dev/ff_17/update_sets';
create directory ff18 as  '/home/oracle/dev/ff_18/update_sets';
create directory ff19 as  '/home/oracle/dev/ff_19/update_sets';
create directory ff20 as  '/home/oracle/dev/ff_20/update_sets';
create directory ff21 as  '/home/oracle/dev/ff_21/update_sets';
create directory ff22 as  '/home/oracle/dev/ff_22/update_sets';
create directory ff23 as  '/home/oracle/dev/ff_23/update_sets';
create directory ff24 as  '/home/oracle/dev/ff_24/update_sets';
create directory ff25 as  '/home/oracle/dev/ff_25/update_sets';
create directory ff26 as  '/home/oracle/dev/ff_26/update_sets';
create directory ff27 as  '/home/oracle/dev/ff_27/update_sets';
create directory ff28 as  '/home/oracle/dev/ff_28/update_sets';
create directory ff29 as  '/home/oracle/dev/ff_29/update_sets';
create directory ff30 as  '/home/oracle/dev/ff_30/update_sets';
create directory ff31 as  '/home/oracle/dev/ff_31/update_sets';
create directory ff32 as  '/home/oracle/dev/ff_32/update_sets';

drop table temp_l_et_${SETNUM};
create table temp_l_et_${SETNUM}(
    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 (
ff1:'lineitem.tbl.u${SETNUM}.1',
ff2:'lineitem.tbl.u${SETNUM}.2',
ff3:'lineitem.tbl.u${SETNUM}.3',
ff4:'lineitem.tbl.u${SETNUM}.4',
ff5:'lineitem.tbl.u${SETNUM}.5',
ff6:'lineitem.tbl.u${SETNUM}.6',
ff7:'lineitem.tbl.u${SETNUM}.7',
ff8:'lineitem.tbl.u${SETNUM}.8',
ff9:'lineitem.tbl.u${SETNUM}.9',
ff10:'lineitem.tbl.u${SETNUM}.10',
ff11:'lineitem.tbl.u${SETNUM}.11',
ff12:'lineitem.tbl.u${SETNUM}.12',
ff13:'lineitem.tbl.u${SETNUM}.13',
ff14:'lineitem.tbl.u${SETNUM}.14',
ff15:'lineitem.tbl.u${SETNUM}.15',
ff16:'lineitem.tbl.u${SETNUM}.16',
ff17:'lineitem.tbl.u${SETNUM}.17',
ff18:'lineitem.tbl.u${SETNUM}.18',
ff19:'lineitem.tbl.u${SETNUM}.19',
ff20:'lineitem.tbl.u${SETNUM}.20',
ff21:'lineitem.tbl.u${SETNUM}.21',
ff22:'lineitem.tbl.u${SETNUM}.22',
ff23:'lineitem.tbl.u${SETNUM}.23',
ff24:'lineitem.tbl.u${SETNUM}.24',
ff25:'lineitem.tbl.u${SETNUM}.25',
ff26:'lineitem.tbl.u${SETNUM}.26',
ff27:'lineitem.tbl.u${SETNUM}.27',
ff28:'lineitem.tbl.u${SETNUM}.28',
ff29:'lineitem.tbl.u${SETNUM}.29',
ff30:'lineitem.tbl.u${SETNUM}.30',
ff31:'lineitem.tbl.u${SETNUM}.31',
ff32:'lineitem.tbl.u${SETNUM}.32'
))
reject limit unlimited parallel ${PAR_HINT};

drop table temp_o_et_${SETNUM};
create table temp_o_et_${SETNUM}(
    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 (
ff1:'orders.tbl.u${SETNUM}.1',
ff2:'orders.tbl.u${SETNUM}.2',
ff3:'orders.tbl.u${SETNUM}.3',
ff4:'orders.tbl.u${SETNUM}.4',
ff5:'orders.tbl.u${SETNUM}.5',
ff6:'orders.tbl.u${SETNUM}.6',
ff7:'orders.tbl.u${SETNUM}.7',
ff8:'orders.tbl.u${SETNUM}.8',
ff9:'orders.tbl.u${SETNUM}.9',
ff10:'orders.tbl.u${SETNUM}.10',
ff11:'orders.tbl.u${SETNUM}.11',
ff12:'orders.tbl.u${SETNUM}.12',
ff13:'orders.tbl.u${SETNUM}.13',
ff14:'orders.tbl.u${SETNUM}.14',
ff15:'orders.tbl.u${SETNUM}.15',
ff16:'orders.tbl.u${SETNUM}.16',
ff17:'orders.tbl.u${SETNUM}.17',
ff18:'orders.tbl.u${SETNUM}.18',
ff19:'orders.tbl.u${SETNUM}.19',
ff20:'orders.tbl.u${SETNUM}.20',
ff21:'orders.tbl.u${SETNUM}.21',
ff22:'orders.tbl.u${SETNUM}.22',
ff23:'orders.tbl.u${SETNUM}.23',
ff24:'orders.tbl.u${SETNUM}.24',
ff25:'orders.tbl.u${SETNUM}.25',
ff26:'orders.tbl.u${SETNUM}.26',
ff27:'orders.tbl.u${SETNUM}.27',
ff28:'orders.tbl.u${SETNUM}.28',
ff29:'orders.tbl.u${SETNUM}.29',
ff30:'orders.tbl.u${SETNUM}.30',
ff31:'orders.tbl.u${SETNUM}.31',
ff32:'orders.tbl.u${SETNUM}.32'
))
reject limit unlimited parallel ${PAR_HINT};

drop table temp_okey_et_${SETNUM};
create table temp_okey_et_${SETNUM}(
    t_orderkey          number
)
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 (
ff1:'delete.u${SETNUM}.1',
ff2:'delete.u${SETNUM}.2',
ff3:'delete.u${SETNUM}.3',
ff4:'delete.u${SETNUM}.4',
ff5:'delete.u${SETNUM}.5',
ff6:'delete.u${SETNUM}.6',
ff7:'delete.u${SETNUM}.7',
ff8:'delete.u${SETNUM}.8',
ff9:'delete.u${SETNUM}.9',
ff10:'delete.u${SETNUM}.10',
ff11:'delete.u${SETNUM}.11',
ff12:'delete.u${SETNUM}.12',
ff13:'delete.u${SETNUM}.13',
ff14:'delete.u${SETNUM}.14',
ff15:'delete.u${SETNUM}.15',
ff16:'delete.u${SETNUM}.16',
ff17:'delete.u${SETNUM}.17',
ff18:'delete.u${SETNUM}.18',
ff19:'delete.u${SETNUM}.19',
ff20:'delete.u${SETNUM}.20',
ff21:'delete.u${SETNUM}.21',
ff22:'delete.u${SETNUM}.22',
ff23:'delete.u${SETNUM}.23',
ff24:'delete.u${SETNUM}.24',
ff25:'delete.u${SETNUM}.25',
ff26:'delete.u${SETNUM}.26',
ff27:'delete.u${SETNUM}.27',
ff28:'delete.u${SETNUM}.28',
ff29:'delete.u${SETNUM}.29',
ff30:'delete.u${SETNUM}.30',
ff31:'delete.u${SETNUM}.31',
ff32:'delete.u${SETNUM}.32'
))
reject limit unlimited  parallel ${PAR_HINT2};
!
----------------------------
create_et_driver.sh
----------------------------
i=0;

while (( i<36 ));do
  (( i=i+1 ))
  /home/oracle/kit/schema/10.0/build/create_et.sh ${i}
done
----------------------------
crts.sh
----------------------------
i=$1;
sqlplus /NOLOG<<!
connect / as sysdba;
drop tablespace tsd${i} including contents;
create tablespace tsd${i} nologging
datafile '/home/oracle/dev/block/lo_${i}' size ${2} reuse extent management dictionary default
storage (initial  25m next 10m maxextents unlimited pctincrease 0);
!

----------------------------
dapop_10gR2_64.sh
----------------------------
#!/bin/bash

sqlplus /NOLOG <<EOF
connect / as sysdba
drop user tpch cascade;
grant DBA
to tpch identified by tpch;
EOF

i=0
while ((i < 64))
do
((i=i+1))
/home/oracle/kit/schema/10.0/build/create_dir.sh $i
done

/home/oracle/kit/schema/10.0/build/create_et_driver.sh

sqlplus /NOLOG <<EOF
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 ff1
access parameters
(
                                                records delimited by newline
                                                nobadfile
                                                nologfile
                                                fields terminated by '|'
                                                missing field values are null
                        )
                        location (
ff1:'lineitem.tbl.1',
ff2:'lineitem.tbl.2',
ff3:'lineitem.tbl.3',
ff4:'lineitem.tbl.4',
ff5:'lineitem.tbl.5',
ff6:'lineitem.tbl.6',
ff7:'lineitem.tbl.7',
ff8:'lineitem.tbl.8',
ff9:'lineitem.tbl.9',
ff10:'lineitem.tbl.10',
ff11:'lineitem.tbl.11',
ff12:'lineitem.tbl.12',
ff13:'lineitem.tbl.13',
ff14:'lineitem.tbl.14',
ff15:'lineitem.tbl.15',
ff16:'lineitem.tbl.16',
ff17:'lineitem.tbl.17',
ff18:'lineitem.tbl.18',
ff19:'lineitem.tbl.19',
ff20:'lineitem.tbl.20',
ff21:'lineitem.tbl.21',
ff22:'lineitem.tbl.22',
ff23:'lineitem.tbl.23',
ff24:'lineitem.tbl.24',
ff25:'lineitem.tbl.25',
ff26:'lineitem.tbl.26',
ff27:'lineitem.tbl.27',
ff28:'lineitem.tbl.28',
ff29:'lineitem.tbl.29',
ff30:'lineitem.tbl.30',
ff31:'lineitem.tbl.31',
ff32:'lineitem.tbl.32',
ff33:'lineitem.tbl.33',
ff34:'lineitem.tbl.34',
ff35:'lineitem.tbl.35',
ff36:'lineitem.tbl.36',
ff37:'lineitem.tbl.37',
ff38:'lineitem.tbl.38',
ff39:'lineitem.tbl.39',
ff40:'lineitem.tbl.40',
ff41:'lineitem.tbl.41',
ff42:'lineitem.tbl.42',
ff43:'lineitem.tbl.43',
ff44:'lineitem.tbl.44',
ff45:'lineitem.tbl.45',
ff46:'lineitem.tbl.46',
ff47:'lineitem.tbl.47',
ff48:'lineitem.tbl.48',
ff49:'lineitem.tbl.49',
ff50:'lineitem.tbl.50',
ff51:'lineitem.tbl.51',
ff52:'lineitem.tbl.52',
ff53:'lineitem.tbl.53',
ff54:'lineitem.tbl.54',
ff55:'lineitem.tbl.55',
ff56:'lineitem.tbl.56',
ff57:'lineitem.tbl.57',
ff58:'lineitem.tbl.58',
ff59:'lineitem.tbl.59',
ff60:'lineitem.tbl.60',
ff61:'lineitem.tbl.61',
ff62:'lineitem.tbl.62',
ff63:'lineitem.tbl.63',
ff64:'lineitem.tbl.64'
))
reject limit unlimited;

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 ff1
access parameters
(
                                                records delimited by newline
                                                nobadfile
                                                nologfile
                                                fields terminated by '|'
                                                missing field values are null
                        )
                        location (
ff1:'orders.tbl.1',
ff2:'orders.tbl.2',
ff3:'orders.tbl.3',
ff4:'orders.tbl.4',
ff5:'orders.tbl.5',
ff6:'orders.tbl.6',
ff7:'orders.tbl.7',
ff8:'orders.tbl.8',
ff9:'orders.tbl.9',
ff10:'orders.tbl.10',
ff11:'orders.tbl.11',
ff12:'orders.tbl.12',
ff13:'orders.tbl.13',
ff14:'orders.tbl.14',
ff15:'orders.tbl.15',
ff16:'orders.tbl.16',
ff17:'orders.tbl.17',
ff18:'orders.tbl.18',
ff19:'orders.tbl.19',
ff20:'orders.tbl.20',
ff21:'orders.tbl.21',
ff22:'orders.tbl.22',
ff23:'orders.tbl.23',
ff24:'orders.tbl.24',
ff25:'orders.tbl.25',
ff26:'orders.tbl.26',
ff27:'orders.tbl.27',
ff28:'orders.tbl.28',
ff29:'orders.tbl.29',
ff30:'orders.tbl.30',
ff31:'orders.tbl.31',
ff32:'orders.tbl.32',
ff33:'orders.tbl.33',
ff34:'orders.tbl.34',
ff35:'orders.tbl.35',
ff36:'orders.tbl.36',
ff37:'orders.tbl.37',
ff38:'orders.tbl.38',
ff39:'orders.tbl.39',
ff40:'orders.tbl.40',
ff41:'orders.tbl.41',
ff42:'orders.tbl.42',
ff43:'orders.tbl.43',
ff44:'orders.tbl.44',
ff45:'orders.tbl.45',
ff46:'orders.tbl.46',
ff47:'orders.tbl.47',
ff48:'orders.tbl.48',
ff49:'orders.tbl.49',
ff50:'orders.tbl.50',
ff51:'orders.tbl.51',
ff52:'orders.tbl.52',
ff53:'orders.tbl.53',
ff54:'orders.tbl.54',
ff55:'orders.tbl.55',
ff56:'orders.tbl.56',
ff57:'orders.tbl.57',
ff58:'orders.tbl.58',
ff59:'orders.tbl.59',
ff60:'orders.tbl.60',
ff61:'orders.tbl.61',
ff62:'orders.tbl.62',
ff63:'orders.tbl.63',
ff64:'orders.tbl.64'
))
reject limit unlimited;

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 ff1
access parameters
(
                                        records delimited by newline
                                        nobadfile
                                        nologfile
                                        fields terminated by '|'
                                        missing field values are null
                    )
                    location (
ff1:'partsupp.tbl.1',
ff2:'partsupp.tbl.2',
ff3:'partsupp.tbl.3',
ff4:'partsupp.tbl.4',
ff5:'partsupp.tbl.5',
ff6:'partsupp.tbl.6',
ff7:'partsupp.tbl.7',
ff8:'partsupp.tbl.8',
ff9:'partsupp.tbl.9',
ff10:'partsupp.tbl.10',
ff11:'partsupp.tbl.11',
ff12:'partsupp.tbl.12',
ff13:'partsupp.tbl.13',
ff14:'partsupp.tbl.14',
ff15:'partsupp.tbl.15',
ff16:'partsupp.tbl.16',
ff17:'partsupp.tbl.17',
ff18:'partsupp.tbl.18',
ff19:'partsupp.tbl.19',
ff20:'partsupp.tbl.20',
ff21:'partsupp.tbl.21',
ff22:'partsupp.tbl.22',
ff23:'partsupp.tbl.23',
ff24:'partsupp.tbl.24',
ff25:'partsupp.tbl.25',
ff26:'partsupp.tbl.26',
ff27:'partsupp.tbl.27',
ff28:'partsupp.tbl.28',
ff29:'partsupp.tbl.29',
ff30:'partsupp.tbl.30',
ff31:'partsupp.tbl.31',
ff32:'partsupp.tbl.32',
ff33:'partsupp.tbl.33',
ff34:'partsupp.tbl.34',
ff35:'partsupp.tbl.35',
ff36:'partsupp.tbl.36',
ff37:'partsupp.tbl.37',
ff38:'partsupp.tbl.38',
ff39:'partsupp.tbl.39',
ff40:'partsupp.tbl.40',
ff41:'partsupp.tbl.41',
ff42:'partsupp.tbl.42',
ff43:'partsupp.tbl.43',
ff44:'partsupp.tbl.44',
ff45:'partsupp.tbl.45',
ff46:'partsupp.tbl.46',
ff47:'partsupp.tbl.47',
ff48:'partsupp.tbl.48',
ff49:'partsupp.tbl.49',
ff50:'partsupp.tbl.50',
ff51:'partsupp.tbl.51',
ff52:'partsupp.tbl.52',
ff53:'partsupp.tbl.53',
ff54:'partsupp.tbl.54',
ff55:'partsupp.tbl.55',
ff56:'partsupp.tbl.56',
ff57:'partsupp.tbl.57',
ff58:'partsupp.tbl.58',
ff59:'partsupp.tbl.59',
ff60:'partsupp.tbl.60',
ff61:'partsupp.tbl.61',
ff62:'partsupp.tbl.62',
ff63:'partsupp.tbl.63',
ff64:'partsupp.tbl.64'
))
reject limit unlimited;

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 ff1
access parameters
(
                                        records delimited by newline
                                        nobadfile
                                        nologfile
                                        fields terminated by '|'
                                        missing field values are null
                    )
                    location (
ff1:'part.tbl.1',
ff2:'part.tbl.2',
ff3:'part.tbl.3',
ff4:'part.tbl.4',
ff5:'part.tbl.5',
ff6:'part.tbl.6',
ff7:'part.tbl.7',
ff8:'part.tbl.8',
ff9:'part.tbl.9',
ff10:'part.tbl.10',
ff11:'part.tbl.11',
ff12:'part.tbl.12',
ff13:'part.tbl.13',
ff14:'part.tbl.14',
ff15:'part.tbl.15',
ff16:'part.tbl.16',
ff17:'part.tbl.17',
ff18:'part.tbl.18',
ff19:'part.tbl.19',
ff20:'part.tbl.20',
ff21:'part.tbl.21',
ff22:'part.tbl.22',
ff23:'part.tbl.23',
ff24:'part.tbl.24',
ff25:'part.tbl.25',
ff26:'part.tbl.26',
ff27:'part.tbl.27',
ff28:'part.tbl.28',
ff29:'part.tbl.29',
ff30:'part.tbl.30',
ff31:'part.tbl.31',
ff32:'part.tbl.32',
ff33:'part.tbl.33',
ff34:'part.tbl.34',
ff35:'part.tbl.35',
ff36:'part.tbl.36',
ff37:'part.tbl.37',
ff38:'part.tbl.38',
ff39:'part.tbl.39',
ff40:'part.tbl.40',
ff41:'part.tbl.41',
ff42:'part.tbl.42',
ff43:'part.tbl.43',
ff44:'part.tbl.44',
ff45:'part.tbl.45',
ff46:'part.tbl.46',
ff47:'part.tbl.47',
ff48:'part.tbl.48',
ff49:'part.tbl.49',
ff50:'part.tbl.50',
ff51:'part.tbl.51',
ff52:'part.tbl.52',
ff53:'part.tbl.53',
ff54:'part.tbl.54',
ff55:'part.tbl.55',
ff56:'part.tbl.56',
ff57:'part.tbl.57',
ff58:'part.tbl.58',
ff59:'part.tbl.59',
ff60:'part.tbl.60',
ff61:'part.tbl.61',
ff62:'part.tbl.62',
ff63:'part.tbl.63',
ff64:'part.tbl.64'
))
reject limit unlimited;

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 ff1
access parameters
(
                                                records delimited by newline
                                                nobadfile
                                                nologfile
                                                fields terminated by '|'
                                                missing field values are null
                        )
                        location (
ff1:'customer.tbl.1',
ff2:'customer.tbl.2',
ff3:'customer.tbl.3',
ff4:'customer.tbl.4',
ff5:'customer.tbl.5',
ff6:'customer.tbl.6',
ff7:'customer.tbl.7',
ff8:'customer.tbl.8',
ff9:'customer.tbl.9',
ff10:'customer.tbl.10',
ff11:'customer.tbl.11',
ff12:'customer.tbl.12',
ff13:'customer.tbl.13',
ff14:'customer.tbl.14',
ff15:'customer.tbl.15',
ff16:'customer.tbl.16',
ff17:'customer.tbl.17',
ff18:'customer.tbl.18',
ff19:'customer.tbl.19',
ff20:'customer.tbl.20',
ff21:'customer.tbl.21',
ff22:'customer.tbl.22',
ff23:'customer.tbl.23',
ff24:'customer.tbl.24',
ff25:'customer.tbl.25',
ff26:'customer.tbl.26',
ff27:'customer.tbl.27',
ff28:'customer.tbl.28',
ff29:'customer.tbl.29',
ff30:'customer.tbl.30',
ff31:'customer.tbl.31',
ff32:'customer.tbl.32',
ff33:'customer.tbl.33',
ff34:'customer.tbl.34',
ff35:'customer.tbl.35',
ff36:'customer.tbl.36',
ff37:'customer.tbl.37',
ff38:'customer.tbl.38',
ff39:'customer.tbl.39',
ff40:'customer.tbl.40',
ff41:'customer.tbl.41',
ff42:'customer.tbl.42',
ff43:'customer.tbl.43',
ff44:'customer.tbl.44',
ff45:'customer.tbl.45',
ff46:'customer.tbl.46',
ff47:'customer.tbl.47',
ff48:'customer.tbl.48',
ff49:'customer.tbl.49',
ff50:'customer.tbl.50',
ff51:'customer.tbl.51',
ff52:'customer.tbl.52',
ff53:'customer.tbl.53',
ff54:'customer.tbl.54',
ff55:'customer.tbl.55',
ff56:'customer.tbl.56',
ff57:'customer.tbl.57',
ff58:'customer.tbl.58',
ff59:'customer.tbl.59',
ff60:'customer.tbl.60',
ff61:'customer.tbl.61',
ff62:'customer.tbl.62',
ff63:'customer.tbl.63',
ff64:'customer.tbl.64'
))
reject limit unlimited;

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 ff1
access parameters
(
                                                records delimited by newline
                                                nobadfile
                                                nologfile
                                                fields terminated by '|'
                                                missing field values are null
                        )
                        location (
ff1:'supplier.tbl.1',
ff2:'supplier.tbl.2',
ff3:'supplier.tbl.3',
ff4:'supplier.tbl.4',
ff5:'supplier.tbl.5',
ff6:'supplier.tbl.6',
ff7:'supplier.tbl.7',
ff8:'supplier.tbl.8',
ff9:'supplier.tbl.9',
ff10:'supplier.tbl.10',
ff11:'supplier.tbl.11',
ff12:'supplier.tbl.12',
ff13:'supplier.tbl.13',
ff14:'supplier.tbl.14',
ff15:'supplier.tbl.15',
ff16:'supplier.tbl.16',
ff17:'supplier.tbl.17',
ff18:'supplier.tbl.18',
ff19:'supplier.tbl.19',
ff20:'supplier.tbl.20',
ff21:'supplier.tbl.21',
ff22:'supplier.tbl.22',
ff23:'supplier.tbl.23',
ff24:'supplier.tbl.24',
ff25:'supplier.tbl.25',
ff26:'supplier.tbl.26',
ff27:'supplier.tbl.27',
ff28:'supplier.tbl.28',
ff29:'supplier.tbl.29',
ff30:'supplier.tbl.30',
ff31:'supplier.tbl.31',
ff32:'supplier.tbl.32',
ff33:'supplier.tbl.33',
ff34:'supplier.tbl.34',
ff35:'supplier.tbl.35',
ff36:'supplier.tbl.36',
ff37:'supplier.tbl.37',
ff38:'supplier.tbl.38',
ff39:'supplier.tbl.39',
ff40:'supplier.tbl.40',
ff41:'supplier.tbl.41',
ff42:'supplier.tbl.42',
ff43:'supplier.tbl.43',
ff44:'supplier.tbl.44',
ff45:'supplier.tbl.45',
ff46:'supplier.tbl.46',
ff47:'supplier.tbl.47',
ff48:'supplier.tbl.48',
ff49:'supplier.tbl.49',
ff50:'supplier.tbl.50',
ff51:'supplier.tbl.51',
ff52:'supplier.tbl.52',
ff53:'supplier.tbl.53',
ff54:'supplier.tbl.54',
ff55:'supplier.tbl.55',
ff56:'supplier.tbl.56',
ff57:'supplier.tbl.57',
ff58:'supplier.tbl.58',
ff59:'supplier.tbl.59',
ff60:'supplier.tbl.60',
ff61:'supplier.tbl.61',
ff62:'supplier.tbl.62',
ff63:'supplier.tbl.63',
ff64:'supplier.tbl.64'
                        ))
reject limit unlimited;

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 ff1
access parameters
(
                                        records delimited by newline
                                        nobadfile
                                        nologfile
                                        fields terminated by '|'
                                        missing field values are null
                    )
                    location (
ff1:'nation.tbl'
))
reject limit unlimited;

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 ff1
access parameters
(
                                        records delimited by newline
                                        nobadfile
                                        nologfile
                                        fields terminated by '|'
                                        missing field values are null
                    )
                    location (
ff1:'region.tbl'
))
reject limit unlimited;
drop table l_et2;
create table l_et2(
   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 ff1
access parameters
(
             records delimited by newline
             nobadfile
             nologfile
             fields terminated by '|'
             missing field values are null
       )
       location (ff64:'lineitem.tbl.64_2'))
reject limit unlimited;

drop table o_et2;
create table o_et2(
   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 ff1
access parameters
(
             records delimited by newline
             nobadfile
             nologfile
             fields terminated by '|'
             missing field values are null
       )
       location (ff64:'orders.tbl.64_2'))
reject limit unlimited;

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;
alter table l_et2 parallel;
alter table o_et2 parallel;

alter user tpch default tablespace ts_default;
alter user tpch temporary tablespace ts_temp;

@?/rdbms/admin/utlxplan.sql;

set timing on
set echo on
!date
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 32m next 4m freelist groups 8 freelists 99)
compress
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 64
(
partition lp1 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp2 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp3 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp4 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp5 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp6 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp7 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp8 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp9 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp10 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp11 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp12 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp13 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp14 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp15 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp16 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp17 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp18 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp19 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp20 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp21 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp22 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp23 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp24 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp25 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp26 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp27 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp28 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp29 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp30 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp31 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp32 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp33 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp34 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp35 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp36 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp37 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp38 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp39 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp40 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp41 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp42 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp43 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp44 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp45 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp46 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp47 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp48 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp49 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp50 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp51 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp52 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp53 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp54 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp55 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp56 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp57 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp58 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp59 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp60 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp61 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp62 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp63 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp64 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp65 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp66 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp67 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp68 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp69 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp70 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp71 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp72 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp73 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp74 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp75 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp76 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
```

```
8,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,t
s_lo63,ts_lo64),
partition lp77 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp78 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp79 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp80 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp81 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp82 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp83 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition lp84 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_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 order by l_orderkey;
!date

!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 8m next 4m freelist groups 8 freelists 99)
compress
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 64
(
partition op1 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op2 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op3 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op4 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op5 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op6 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op7 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op8 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op9 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op10 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op11 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op12 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op13 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op14 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op15 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op16 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op17 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op18 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op19 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op20 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op21 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op22 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op23 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op24 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op25 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op26 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op27 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op28 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op29 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op30 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op31 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op32 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op33 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op34 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op35 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op36 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op37 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op38 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op39 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op40 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op41 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op42 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op43 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op44 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op45 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op46 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op47 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op48 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op49 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op50 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op51 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op52 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op53 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op54 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op55 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op56 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
```

```
8,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,t
s_lo63,ts_lo64),
partition op57 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op58 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op59 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op60 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op61 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op62 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op63 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op64 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op65 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op66 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op67 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op68 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op69 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op70 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op71 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op72 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op73 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op74 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op75 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op76 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op77 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op78 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op79 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op80 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op81 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op82 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op83 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64),
partition op84 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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_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 order by o_orderkey;
!date

rem drop table partsupp;
create table partsupp(
    ps_partkey           NOT NULL,
    ps_suppkey           NOT NULL,
    ps_supplycost        NOT NULL,
    ps_availqty          ,
    ps_comment
)
partition by hash(ps_partkey)
partitions 64
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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
storage (initial 64m next 4m freelist groups 8 freelists 99)
compress
parallel
nologging
as select
    ps_partkey           ,
    ps_suppkey           ,
    ps_supplycost        ,
    ps_availqty          ,
    ps_comment
from ps_et;
!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
storage (initial 4m next 4m freelist groups 8 freelists 99)
compress
parallel
nologging
partition by hash (c_custkey)
partitions 64
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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
as select
    c_custkey            ,
    c_mktsegment         ,
    c_nationkey          ,
    c_name               ,
    c_address            ,
    c_phone              ,
    c_acctbal            ,
    c_comment
from c_et;
!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
storage (initial 16m next 4m  freelist groups 8 freelists 99)
compress
parallel
nologging
partition by hash (p_partkey)
partitions 64
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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
as select
    p_partkey            ,
    p_type               ,
    p_size               ,
    p_brand              ,
    p_name               ,
    p_container          ,
    p_mfgr               ,
    p_retailprice        ,
    p_comment
from p_et;
!date

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
storage (initial 16m next 4m freelist groups 8 freelists 99)
compress
parallel
nologging
partition by hash (s_suppkey)
partitions 64
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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
```

```
8,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,t
s_lo63,ts_lo64)
as select
    s_suppkey           ,
    s_nationkey         ,
    s_comment           ,
    s_name              ,
    s_address           ,
    s_phone             ,
    s_acctbal
from s_et;
!date

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

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

!date

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_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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
pctfree 2
initrans 10
storage (initial 4m next 4m  freelist groups 8 freelists 99)
parallel
compute statistics
nologging;

!date

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_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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
pctfree 2
initrans 10
storage (initial 4m next 4m freelist groups 8 freelists 99 )
parallel
compute statistics
nologging;

!date

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_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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
pctfree 2
initrans 10
storage (initial 4m next 4m freelist groups 8 freelists 99)
parallel
compute statistics
nologging;

create unique index i_ps_partkey_suppkey
on partsupp (ps_partkey,ps_suppkey)  global partition by hash (ps_partkey)
partitions 64
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_l
o14,ts_lo15,ts_lo16,ts_lo17,ts_lo18,ts_lo19,ts_lo20,ts_lo21,ts_lo22,ts_lo23,ts_lo24,ts_lo25,ts_lo
26,ts_lo27,ts_lo28,ts_lo29,ts_lo30,ts_lo31,ts_lo32,ts_lo33,ts_lo34,ts_lo35,ts_lo36,ts_lo37,ts_lo3
8,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,t
s_lo63,ts_lo64)
pctfree 2
initrans 10
storage (initial 4m next 4m freelist groups 8 freelists 99)
parallel
compute statistics
nologging;

alter session force parallel dml parallel (degree 64);

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 l_et2);

insert into orders (
   select
    o_orderdate         ,
    o_orderkey          ,
    o_custkey           ,
    o_orderpriority     ,
    o_shippriority      ,
    o_clerk             ,
    o_orderstatus       ,
    o_totalprice        ,
    o_comment
from o_et2);
commit;
drop table o_et2;
drop table l_et2;

!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;
exec dbms_scheduler.disable('GATHER_STATS_JOB');
exec dbms_scheduler.disable('AUTO_SPACE_ADVISOR_JOB');
exec dbms_scheduler.disable('AUTO_TASKS_JOB_CLASS');
alter system switch logfile;
!date
EOF

/home/oracle/kit/schema/10.0/build/preallocate_drv.sh i_l_orderkey 64 4m
/home/oracle/kit/schema/10.0/build/preallocate_drv.sh i_o_orderkey 64 4m

----------------------------
dbcre_10gR2_cclass.sh
----------------------------
#!/bin/ksh
echo "database creation"
date;

sqlplus  /NOLOG <<!
connect /as sysdba

startup pfile = /home/oracle/oracle/product/10.2.0/db_1/dbs/init_build.ora nomount;
create database
  controlfile reuse
  logfile group 1 ('/home/oracle/dev/block/log_1')  size 1023m reuse,
          group 2 ('/home/oracle/dev/block/log_2')  size 1023m reuse
  datafile '/home/oracle/dev/block/sys1'  size 1023m reuse
```

```
 sysaux datafile '/home/oracle/dev/block/sysaux1' size 1023m reuse
 undo tablespace ts_undo1
   datafile '/home/oracle/dev/block/undo_1' size 1023m reuse
 default temporary tablespace ts_temp
   tempfile '/home/oracle/dev/block/t_1' size 7000m reuse
   extent management local uniform size 10m
 maxdatafiles 2000
 maxinstances 8;

create undo tablespace ts_undo2 datafile '/home/oracle/dev/block/undo_2' size 1023m reuse;
create undo tablespace ts_undo3 datafile '/home/oracle/dev/block/undo_3' size 1023m reuse;
create undo tablespace ts_undo4 datafile '/home/oracle/dev/block/undo_4' size 1023m reuse;
create undo tablespace ts_undo5 datafile '/home/oracle/dev/block/undo_5' size 1023m reuse;
create undo tablespace ts_undo6 datafile '/home/oracle/dev/block/undo_6' size 1023m reuse;
create undo tablespace ts_undo7 datafile '/home/oracle/dev/block/undo_7' size 1023m reuse;
create undo tablespace ts_undo8 datafile '/home/oracle/dev/block/undo_8' size 1023m reuse;

alter database add logfile thread 2  group 3 ('/home/oracle/dev/block/log_3') size 1023m reuse;
alter database add logfile thread 2  group 4 ('/home/oracle/dev/block/log_4') size 1023m reuse;
alter database add logfile thread 3  group 5 ('/home/oracle/dev/block/log_5') size 1023m reuse;
alter database add logfile thread 3  group 6 ('/home/oracle/dev/block/log_6') size 1023m reuse;
alter database add logfile thread 4  group 7 ('/home/oracle/dev/block/log_7') size 1023m reuse;
alter database add logfile thread 4  group 8 ('/home/oracle/dev/block/log_8') size 1023m reuse;
alter database add logfile thread 5  group 9 ('/home/oracle/dev/block/log_9') size 1023m reuse;
alter database add logfile thread 5  group 10 ('/home/oracle/dev/block/log_10') size 1023m reuse;
alter database add logfile thread 6  group 11 ('/home/oracle/dev/block/log_11') size 1023m reuse;
alter database add logfile thread 6  group 12 ('/home/oracle/dev/block/log_12') size 1023m reuse;
alter database add logfile thread 7  group 13 ('/home/oracle/dev/block/log_13') size 1023m reuse;
alter database add logfile thread 7  group 14 ('/home/oracle/dev/block/log_14') size 1023m reuse;
alter database add logfile thread 8  group 15 ('/home/oracle/dev/block/log_15') size 1023m reuse;
alter database add logfile thread 8  group 16 ('/home/oracle/dev/block/log_16') size 1023m reuse;

alter database enable public thread 2;
alter database enable public thread 3;
alter database enable public thread 4;
alter database enable public thread 5;
alter database enable public thread 6;
alter database enable public thread 7;
alter database enable public thread 8;

set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catproc.sql;
@?/rdbms/admin/catclust.sql;
connect system/manager
@?/sqlplus/admin/pupbld.sql;
spool off
!
echo "end of database creation"
```

```
date
----------------------------
preallocate_drv.sh
----------------------------
index=$1
num=$2
size=$3
((i=0))
while ((i<num));do
  ((i=i+1))
  /home/oracle/kit/schema/10.0/build/preallocate.sh $index $size
done
----------------------------
tscre_10gR2.sh
----------------------------
#!/bin/ksh
echo "START: tablespace creation"
date;

sqlplus  /NOLOG <<!
connect /as sysdba

create tablespace ts_default
datafile '/home/oracle/dev/block/default' size 512m reuse
extent management local autoallocate;
!
i=0
while [ $i -lt 64 ]
do
        i=`expr $i + 1`
        ./crets.sh  ts_lo$i /home/oracle/dev/block/lo_$i 7000m  &
                        sleep 6
done

wait;


i=1
while [ $i -lt 64 ]
do
        i=`expr $i + 1`
        ./addtts.sh  ts_temp /home/oracle/dev/block/t_$i 7000m &
                        sleep 3
done

wait;


echo "END: tablespace  creation"
date;
```

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_ATTR_USERNAME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_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_SESSION,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),SQLT_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),SQLT_INT);
  OCIbbname(curr,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_INT);
  OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQLT_INT);
  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),SQLT_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),SQLT_FLT);
  OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_eprice),
           SQLT_FLT);
  OCIbbname(curr,l_neweprice_bp,errhp,":l_neweprice",ADR(l_neweprice),
           SIZ(l_neweprice),SQLT_FLT);

  OCIbbname(curr,o_tprice_bp,errhp,":o_tprice",ADR(o_tprice),SIZ(o_tprice),
           SQLT_FLT);
  OCIbbname(curr,o_newtprice_bp,errhp,":o_newtprice",ADR(o_newtprice),
           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_neweprice),
           SIZ(l_neweprice),SQLT_FLT);
  OCIbbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_newquan),
           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),SQLT_INT);

  OCIbbname(cure2,o_newtprice2_bp,errhp,":o_newtprice",ADR(o_newtprice),
           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_DEFAULT)) !=
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=10"

#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 */
----------------------------
atrans.sql
----------------------------
Rem
Rem $Header: atrans.sql 07-aug-99.21:27:13 mpoess Exp $
Rem
Rem atrans.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      atrans.sql - <one-line expansion of the name>
Rem
Rem   DESCRIPTION
Rem       Creates ACID Transaction Package for TPC-D benchmark.
Rem       Asks user to input values for o_key, delta and output file.
Rem
Rem    NOTES
Rem     <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/07/99 - Creation
Rem    mpoess      08/07/99 - Created
Rem


set serverout on;
set termout on;
set echo on;

CREATE OR REPLACE PACKAGE d_atrans
IS
PROCEDURE doatrans
(
                l_key                           IN OUT integer,
                o_key                           IN OUT integer,
                delta                           IN OUT integer,
                l_pkey          IN OUT integer,
                l_skey          IN OUT integer,
                l_quan          IN OUT integer,
                l_newquan       IN OUT integer,
                l_tax           IN OUT number,
                l_disc          IN OUT number,
                l_eprice        IN OUT number,
                l_neweprice     IN OUT number,
                o_tprice        IN OUT number,
                o_newtprice     IN OUT number,
                rprice                          IN OUT number,
                cost                            IN OUT number
);
END;
/

CREATE OR REPLACE PACKAGE BODY d_atrans
IS
PROCEDURE doatrans
(
                l_key                           IN OUT integer,
                o_key                           IN OUT integer,
                delta                           IN OUT integer,
                l_pkey          IN OUT integer,
                l_skey          IN OUT integer,
                l_quan          IN OUT integer,
                l_newquan       IN OUT integer,
                l_tax           IN OUT number,
                l_disc          IN OUT number,
                l_eprice        IN OUT number,
                l_neweprice     IN OUT number,
                o_tprice        IN OUT number,
                o_newtprice     IN OUT number,
                rprice                          IN OUT number,
                cost                            IN OUT number
)
```

```
IS
    ototal number;
    not_serializable EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-8177);
BEGIN
 LOOP BEGIN

  select o_totalprice
            into o_tprice
            from orders
            where o_orderkey = o_key;

  select l_quantity, l_extendedprice, l_partkey, l_suppkey, l_tax, l_discount
    into l_quan, l_eprice, l_pkey, l_skey, l_tax, l_disc
    from lineitem
    where l_orderkey = o_key
    and   l_linenumber = l_key;

  ototal := o_tprice - trunc((trunc((l_eprice * (1.0-l_disc)),2) * (1.0+l_tax)),2);
  rprice := trunc((l_eprice/l_quan), 2);
  cost := trunc((rprice * delta), 2);
  l_neweprice := l_eprice + cost;
  o_newtprice := trunc((l_neweprice * (1.0 - l_disc)), 2);
  o_newtprice := ototal + trunc((o_newtprice * (1.0 + l_tax)), 2);
  l_newquan := l_quan + delta;

  update lineitem
    set l_extendedprice = l_neweprice,
    l_quantity = l_newquan
    where l_orderkey = o_key
    and l_linenumber = l_key;

  update orders
    set o_totalprice = o_newtprice
    where o_orderkey = o_key;

   insert into history (h_p_key, h_s_key, h_o_key, h_l_key, h_delta, h_date_t)
    values (l_pkey, l_skey, o_key, l_key, delta, sysdate);

  EXIT;

 EXCEPTION
  WHEN not_serializable THEN
   ROLLBACK;
 END;

 END LOOP;

END doatrans;
END;
/
exit;
----------------------------
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;
!
----------------------------
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

----------------------------
d_hist.sql
----------------------------
Rem
Rem $Header: d_hist.sql 07-aug-99.21:33:08 mpoess Exp $
Rem
Rem d_hist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
Rem    NAME
Rem      d_hist.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem       Creates a history table for ACID test purpose.
Rem
Rem    NOTES
Rem     <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess      08/07/99 - Creation
Rem    mpoess      08/07/99 - Created
Rem


set termout on;
set serverout on;
set echo on;

drop table history;

create table history
(
            h_p_key         number,
            h_s_key         number,
                h_o_key number,
                h_l_key         number,
                h_delta number,
                h_date_t date
);

exit;
----------------------------
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
----------------------------
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 */

----------------------------
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=ssh

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=ssh

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=/home/oracle/kit/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
scp $KEYFILE ${HOST}:$KEYFILE

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

$PROG 1 2 1 0 i$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
----------------------------
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=ssh

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=/home/oracle/kit/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
scp $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=ssh

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=/home/oracle/kit/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
scp $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=ssh

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=/home/oracle/kit/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
scp $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_ATTR_USERNAME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_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_SESSION,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),SQLT_INT);
  OCIbbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_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: Qualification query text and output

```
----------------------------
1.log
----------------------------
Begin Execution at Mon Aug  6 11:43:29 2007



-- 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 6.89 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:36 2007


Stream Started at 1186418609.87
Stream Ended at 1186418616.75
Stream Processed in 6.89 seconds


SQL statements processed: 1
----------------------------
1.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:29 2007


Query : Execution Time: 6.89 started 1186418609.87 ended 1186418616.75

Ended Executing this Stream at Mon Aug  6 11:43:36 2007


Stream Started at 1186418609.87
Stream Ended at 1186418616.75
Stream Processed in 6.89 seconds

----------------------------
1.sql
----------------------------
```

```
-- 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;

----------------------------
10.log
----------------------------
Begin Execution at Mon Aug  6 11:43:46 2007



-- 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
sits. slyly regular requests sleep alongside of the regular inst
143347.00          Customer#000143347       721002.69
2557.47            EGYPT
```

```
1aReFYv,Kw4                          14-742-935-3718
ggle carefully enticing requests. final deposits use bold, bold pinto beans. ironic, idle re
60838.00              Customer#000060838       679127.31
2454.77               BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE            12-913-494-9813
 need to boost against the slyly regular account
101998.00             Customer#000101998       637029.57
3790.89               UNITED KINGDOM
01c9CILnNtfOQYmZj                       33-593-865-6378
ress foxes wake slyly after the bold excuses. ironic platelets are furiously carefully bold
theodolites
125341.00             Customer#000125341       633508.09
4983.51               GERMANY
S29ODD6bceU8QSuuEJznkNaK                17-582-695-5962
arefully even depths. blithely even excuses sleep furiously. foxes use except the dependencies. ca
25501.00              Customer#000025501       620269.78
7725.04               ETHIOPIA
  W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ     15-874-808-6793
he pending instructions wake carefully at the pinto beans. regular, final instructions along the
slyly fina
115831.00             Customer#000115831       596423.87
5098.10               FRANCE
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-3788
l somas sleep. furiously final deposits wake blithely regular pinto b
84223.00              Customer#000084223       594998.02
528.65               UNITED KINGDOM
nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA      33-442-824-8191
 slyly final deposits haggle regular, pending dependencies. pending escapades wake
54289.00              Customer#000054289       585603.39
5583.02               IRAN
vXCxoCsU0Bad5JQI ,oobkZ                 20-834-292-4707
ely special foxes are quickly finally ironic p
39922.00              Customer#000039922       584878.11
7321.11               GERMANY
Zgy4s50l2GKN4pLDPBU8m342gIw6R           17-147-757-8036
y final requests. furiously final foxes cajole blithely special platelets. f
6226.00               Customer#000006226       576783.76
2230.09               UNITED KINGDOM
8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g,       33-657-701-3391
ending platelets along the express deposits cajole carefully final
922.00               Customer#000000922       576767.53
3869.25               GERMANY
Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq         17-945-916-9648
luffily fluffy deposits. packages c
147946.00             Customer#000147946       576455.13
2030.13               ALGERIA
iANyZHjqhyy7Ajah0pTrYyhJ                10-886-956-3143
ithely ironic deposits haggle blithely ironic requests. quickly regu
115640.00             Customer#000115640       569341.19
6436.10               ARGENTINA
Vtgfia9qI 7EpHgecU1X                    11-411-543-4901
ost slyly along the patterns; pinto be
73606.00              Customer#000073606       568656.86
1785.67               JAPAN
xuR0Tro5yChDfOCrjkd2ol                  22-437-653-6966
he furiously regular ideas. slowly
110246.00             Customer#000110246       566842.98
7763.35               VIETNAM
7KzflgX MDOq7sOkI                       31-943-426-9837
egular deposits serve blithely above the fl
142549.00             Customer#000142549       563537.24
5085.99               INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9          19-955-562-2398
sleep pending courts. ironic deposits against the carefully unusual platelets cajole carefully
express accounts.
146149.00             Customer#000146149       557254.99
1791.55               ROMANIA
s87fvzFQpU                              29-744-164-6487
 of the slyly silent accounts. quickly final accounts across the
52528.00              Customer#000052528       556397.35
551.79               ARGENTINA
NFztyTOR10UOJ                           11-208-192-3205
 deposits hinder. blithely pending asymptotes breach slyly regular re
23431.00              Customer#000023431       554269.54
3381.86               ROMANIA
HgiV0phqhaIa9aydNoIlb                   29-915-458-2654
nusual, even instructions: furiously stealthy n


20 rows processed.
Query Processed in 1.54 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:48 2007


Stream Started at 1186418626.84
Stream Ended at 1186418628.38
Stream Processed in 1.54 seconds


SQL statements processed: 1
----------------------------
10.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:46 2007


Query : Execution Time: 1.54 started 1186418626.84 ended 1186418628.38

Ended Executing this Stream at Mon Aug  6 11:43:48 2007


Stream Started at 1186418626.84
Stream Ended at 1186418628.38
Stream Processed in 1.54 seconds

----------------------------
10.sql
----------------------------
-- 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;

----------------------------
11.log
----------------------------
Begin Execution at Mon Aug  6 11:43:48 2007



-- 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
9403.00              15451755.62


:
:
:
:
:
:
<< Lines deleted >>
:
:
:
:
105235.00            7897829.94
77207.00             7897752.72
96712.00             7897575.27
10157.00             7897046.25
171154.00            7896814.50
79373.00             7896186.00
113808.00            7893353.88
27901.00             7892952.00
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 0.39 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:48 2007


Stream Started at 1186418628.41
Stream Ended at 1186418628.80
Stream Processed in 0.39 seconds


SQL statements processed: 1
----------------------------
11.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:48 2007


Query : Execution Time: 0.39 started 1186418628.41 ended 1186418628.80

Ended Executing this Stream at Mon Aug  6 11:43:48 2007


Stream Started at 1186418628.41
Stream Ended at 1186418628.80
Stream Processed in 0.39 seconds

----------------------------
11.sql
----------------------------
-- 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;

----------------------------
12.log
----------------------------
Begin Execution at Mon Aug  6 11:43:48 2007



-- 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 0.80 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:49 2007


Stream Started at 1186418628.83
Stream Ended at 1186418629.63
Stream Processed in 0.80 seconds


SQL statements processed: 1
----------------------------
12.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:48 2007


Query : Execution Time: 0.80 started 1186418628.83 ended 1186418629.63

Ended Executing this Stream at Mon Aug  6 11:43:49 2007


Stream Started at 1186418628.83
Stream Ended at 1186418629.63
Stream Processed in 0.80 seconds

----------------------------
12.sql
----------------------------
-- 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;
----------------------------
13.log
----------------------------
Begin Execution at Mon Aug  6 11:43:49 2007



-- 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                50005.00
9.00                6641.00
10.00                6532.00
11.00                6014.00
8.00                5937.00
12.00                5639.00
13.00                5024.00
19.00                4793.00
7.00                4687.00
17.00                4587.00
18.00                4529.00
20.00                4516.00
15.00                4505.00
14.00                4446.00
16.00                4273.00
21.00                4190.00
22.00                3623.00
6.00                3265.00
23.00                3225.00
24.00                2742.00
25.00                2086.00
5.00                1948.00
26.00                1612.00
27.00                1179.00
4.00                1007.00
28.00                893.00
29.00                593.00
3.00                415.00
30.00                376.00
31.00                226.00
32.00                148.00
2.00                134.00
33.00                75.00
34.00                50.00
35.00                37.00
1.00                17.00
36.00                14.00
38.00                5.00
```

```
37.00              5.00
40.00              4.00
41.00              2.00
39.00              1.00


42 rows processed.
Query Processed in 1.82 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:51 2007


Stream Started at 1186418629.67
Stream Ended at 1186418631.49
Stream Processed in 1.82 seconds


SQL statements processed: 1
----------------------------
13.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:49 2007


Query : Execution Time: 1.82 started 1186418629.67 ended 1186418631.49

Ended Executing this Stream at Mon Aug  6 11:43:51 2007


Stream Started at 1186418629.67
Stream Ended at 1186418631.49
Stream Processed in 1.82 seconds

----------------------------
13.sql
----------------------------
-- 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;

----------------------------
14.log
----------------------------
Begin Execution at Mon Aug  6 11:43:51 2007



-- 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 0.12 seconds.
```

```
Ended Executing this Stream at Mon Aug  6 11:43:51 2007


Stream Started at 1186418631.52
Stream Ended at 1186418631.64
Stream Processed in 0.12 seconds


SQL statements processed: 1
----------------------------
14.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:51 2007


Query : Execution Time: 0.12 started 1186418631.52 ended 1186418631.64

Ended Executing this Stream at Mon Aug  6 11:43:51 2007


Stream Started at 1186418631.52
Stream Ended at 1186418631.64
Stream Processed in 0.12 seconds

----------------------------
14.sql
----------------------------
-- 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;
----------------------------
15.log
----------------------------
Begin Execution at Mon Aug  6 11:43:51 2007



-- 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 0.58 seconds.




drop view revenue0
Query Processed in 0.02 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:52 2007


Stream Started at 1186418631.67
Stream Ended at 1186418632.29
Stream Processed in 0.62 seconds


SQL statements processed: 3
----------------------------
15.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:51 2007


Query : Execution Time: 0.02 started 1186418631.67 ended 1186418631.69
Query : Execution Time: 0.58 started 1186418631.69 ended 1186418632.27
Query : Execution Time: 0.02 started 1186418632.27 ended 1186418632.29

Ended Executing this Stream at Mon Aug  6 11:43:52 2007


Stream Started at 1186418631.67
Stream Ended at 1186418632.29
Stream Processed in 0.62 seconds

----------------------------
15.sql
----------------------------
-- 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;


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;

drop view revenue0;

----------------------------
16.log
----------------------------
Begin Execution at Mon Aug  6 11:43:52 2007



-- 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
Brand#21   MEDIUM ANODIZED COPPER    3.00                  24.00
Brand#22   SMALL BRUSHED NICKEL      3.00                  24.00
Brand#22   SMALL BURNISHED BRASS     19.00                 24.00
Brand#25   MEDIUM BURNISHED COPPER   36.00                 24.00
Brand#31   PROMO POLISHED COPPER     36.00                 24.00
Brand#33   LARGE POLISHED TIN        23.00                 24.00
Brand#33   PROMO POLISHED STEEL      14.00                 24.00
Brand#35   PROMO BRUSHED NICKEL      14.00                 24.00
Brand#41   ECONOMY BRUSHED STEEL     9.00                  24.00
Brand#41   ECONOMY POLISHED TIN      19.00                 24.00


:
:
:
:
:
:
<< Lines deleted >>
:
:
:
:
Brand#55   STANDARD POLISHED STEEL   23.00                 4.00
Brand#55   STANDARD POLISHED TIN     9.00                  4.00
Brand#55   STANDARD POLISHED TIN     19.00                 4.00
Brand#55   STANDARD POLISHED TIN     36.00                 4.00
Brand#11   SMALL BRUSHED TIN         19.00                 3.00
Brand#15   LARGE PLATED NICKEL       45.00                 3.00
Brand#15   LARGE POLISHED NICKEL     9.00                  3.00
Brand#21   PROMO BURNISHED STEEL     45.00                 3.00
Brand#22   STANDARD PLATED STEEL     23.00                 3.00
Brand#25   LARGE PLATED STEEL        19.00                 3.00
Brand#32   STANDARD ANODIZED COPPER  23.00                 3.00
Brand#33   SMALL ANODIZED BRASS      9.00                  3.00
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 0.53 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:52 2007


Stream Started at 1186418632.32
Stream Ended at 1186418632.86
Stream Processed in 0.53 seconds
```

```
SQL statements processed: 1
----------------------------
16.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:52 2007


Query : Execution Time: 0.53 started 1186418632.32 ended 1186418632.86

Ended Executing this Stream at Mon Aug  6 11:43:52 2007


Stream Started at 1186418632.32
Stream Ended at 1186418632.86
Stream Processed in 0.53 seconds

----------------------------
16.sql
----------------------------
-- 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;

----------------------------
17.log
----------------------------
Begin Execution at Mon Aug  6 11:43:52 2007



-- 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 1.04 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:53 2007
```

```
Stream Started at 1186418632.89
Stream Ended at 1186418633.93
Stream Processed in 1.04 seconds


SQL statements processed: 1
----------------------------
17.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:52 2007


Query : Execution Time: 1.04 started 1186418632.89 ended 1186418633.93

Ended Executing this Stream at Mon Aug  6 11:43:53 2007


Stream Started at 1186418632.89
Stream Ended at 1186418633.93
Stream Processed in 1.04 seconds

----------------------------
17.sql
----------------------------
-- 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
        );

----------------------------
18.log
----------------------------
Begin Execution at Mon Aug  6 11:43:53 2007



-- 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
499794.58               303.00


:
:
:
:
:
:
<< Lines deleted >>
:
:
:
:
412754.51               302.00
Customer#000149842        149842.00              5156581.00            1994-05-30
411329.35               302.00
Customer#000010129        10129.00               5849444.00            1994-03-21
409129.85               309.00
Customer#000069904        69904.00               1742403.00            1996-10-19
408513.00               305.00
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 4.07 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:58 2007


Stream Started at 1186418633.96
Stream Ended at 1186418638.03
Stream Processed in 4.07 seconds


SQL statements processed: 1
----------------------------
18.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:53 2007


Query : Execution Time: 4.07 started 1186418633.96 ended 1186418638.03

Ended Executing this Stream at Mon Aug  6 11:43:58 2007


Stream Started at 1186418633.96
Stream Ended at 1186418638.03
Stream Processed in 4.07 seconds

----------------------------
18.sql
----------------------------
-- 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;

----------------------------
19.log
----------------------------
Begin Execution at Mon Aug  6 11:43:58 2007



-- 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 1.56 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:59 2007


Stream Started at 1186418638.07
Stream Ended at 1186418639.63
Stream Processed in 1.56 seconds


SQL statements processed: 1
----------------------------
19.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:58 2007
```

```
Query : Execution Time: 1.56 started 1186418638.07 ended 1186418639.63

Ended Executing this Stream at Mon Aug  6 11:43:59 2007


Stream Started at 1186418638.07
Stream Ended at 1186418639.63
Stream Processed in 1.56 seconds

----------------------------
19.sql
----------------------------
-- 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'
        );

----------------------------
2.log
----------------------------
Begin Execution at Mon Aug  6 11:43:36 2007



-- 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
uriously regular requests hag
9937.84         Supplier#000005969     ROMANIA
108438.00         Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa     29-520-692-3537
efully express instructions. regular requests against the slyly fin
9936.22         Supplier#000005250     UNITED KINGDOM
249.00          Manufacturer#4
B3rqp0xbSEim4Mpy2RH J              33-320-228-2957
etect about the furiously final accounts. slyly ironic pinto beans sleep inside the furiously


:
:
:
:
:
:
<< Lines deleted >>
:
:
:
:
ironic requests. furiously final theodolites cajole. final, express packages sleep. quickly reg
7852.45         Supplier#000005864     RUSSIA
8363.00         Manufacturer#4
WCNfBPZeSXh3h,c                    32-454-883-3821
usly unusual pinto beans. brave ideas sleep carefully quickly ironi
7850.66         Supplier#000001518     UNITED KINGDOM
86501.00         Manufacturer#1
ONda3YJiHKJOC                      33-730-383-3892
ifts haggle fluffily pending pai
7843.52         Supplier#000006683     FRANCE
11680.00         Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy         16-464-517-8943
 express, final pinto beans x-ray slyly asymptotes. unusual, unusual


100 rows processed.
Query Processed in 0.21 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:36 2007


Stream Started at 1186418616.79
Stream Ended at 1186418616.99
Stream Processed in 0.21 seconds


SQL statements processed: 1
----------------------------
2.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:36 2007


Query : Execution Time: 0.21 started 1186418616.79 ended 1186418616.99

Ended Executing this Stream at Mon Aug  6 11:43:36 2007


Stream Started at 1186418616.79
Stream Ended at 1186418616.99
Stream Processed in 0.21 seconds

----------------------------
2.sql
----------------------------
-- 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;


----------------------------
20.log
----------------------------
Begin Execution at Mon Aug  6 11:43:59 2007



-- 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,
Supplier#000000688        D fw5ocppmZpYBBIPI718hCihLDZ5KhKX
Supplier#000000710        f19YPvOyb QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736        l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL7
Supplier#000000761        zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
Supplier#000000884        bmhESShejaS
Supplier#000000887        urEaTejH5POADP2ARrf
Supplier#000000935        ij98czM 2KzWe7dDTOxB8sq0UfCdvrX
Supplier#000000975        ,AC e,tBpNwKb5xMUzeohxlRn, hdZJo73gFQF8y
Supplier#000001263        rQWr6nf8ZhB2TAiIDIvo5Io

:
:
:
:
:
:
<< Lines deleted >>
:
:
:
:
Supplier#000009601        51m637bO,Rw5DnHWFUvLacRx9
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 0.42 seconds.


Ended Executing this Stream at Mon Aug  6 11:44:00 2007


Stream Started at 1186418639.66
Stream Ended at 1186418640.08
Stream Processed in 0.42 seconds


SQL statements processed: 1
----------------------------
20.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:59 2007


Query : Execution Time: 0.42 started 1186418639.66 ended 1186418640.08

Ended Executing this Stream at Mon Aug  6 11:44:00 2007


Stream Started at 1186418639.66
Stream Ended at 1186418640.08
Stream Processed in 0.42 seconds

----------------------------
20.sql
----------------------------
-- 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;

----------------------------
21.log
----------------------------
Begin Execution at Mon Aug  6 11:44:00 2007



-- 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
Supplier#000005178       17.00
```

```
Supplier#000008331      17.00
Supplier#000002005      16.00
Supplier#000002095      16.00
Supplier#000005799      16.00
Supplier#000005842      16.00
Supplier#000006450      16.00
Supplier#000006939      16.00
Supplier#000009200      16.00
Supplier#000009727      16.00
Supplier#000000486      15.00
Supplier#000000565      15.00
Supplier#000001046      15.00
Supplier#000001047      15.00
Supplier#000001161      15.00
Supplier#000001336      15.00
Supplier#000001435      15.00
Supplier#000003075      15.00
Supplier#000003335      15.00
Supplier#000005649      15.00
Supplier#000006027      15.00
Supplier#000006795      15.00
Supplier#000006800      15.00
Supplier#000006824      15.00
Supplier#000007131      15.00
Supplier#000007382      15.00
Supplier#000008913      15.00
Supplier#000009787      15.00
Supplier#000000633      14.00
Supplier#000001960      14.00
Supplier#000002323      14.00
Supplier#000002490      14.00
Supplier#000002993      14.00
Supplier#000003101      14.00
Supplier#000004489      14.00
Supplier#000005435      14.00
Supplier#000005583      14.00
Supplier#000005774      14.00
Supplier#000007579      14.00
Supplier#000008180      14.00
Supplier#000008695      14.00
Supplier#000009224      14.00
Supplier#000000357      13.00
Supplier#000000436      13.00
Supplier#000000610      13.00
Supplier#000000788      13.00
Supplier#000000889      13.00
Supplier#000001062      13.00
Supplier#000001498      13.00
Supplier#000002056      13.00
Supplier#000002312      13.00
Supplier#000002344      13.00
Supplier#000002596      13.00
Supplier#000002615      13.00
Supplier#000002978      13.00
Supplier#000003048      13.00
Supplier#000003234      13.00
Supplier#000003727      13.00
Supplier#000003806      13.00
Supplier#000004472      13.00
Supplier#000005236      13.00
Supplier#000005906      13.00
Supplier#000006241      13.00
Supplier#000006326      13.00
Supplier#000006384      13.00
Supplier#000006394      13.00
Supplier#000006624      13.00
Supplier#000006629      13.00
Supplier#000006682      13.00
Supplier#000006737      13.00
Supplier#000006825      13.00
Supplier#000007021      13.00
Supplier#000007417      13.00
Supplier#000007497      13.00
Supplier#000007602      13.00
Supplier#000008134      13.00
Supplier#000008234      13.00
Supplier#000009435      13.00
Supplier#000009436      13.00
Supplier#000009564      13.00
Supplier#000009896      13.00
Supplier#000000379      12.00
Supplier#000000673      12.00
Supplier#000000762      12.00
Supplier#000000811      12.00
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 3.71 seconds.


Ended Executing this Stream at Mon Aug  6 11:44:03 2007


Stream Started at 1186418640.12
Stream Ended at 1186418643.83
Stream Processed in 3.71 seconds


SQL statements processed: 1
----------------------------
21.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:44:00 2007


Query : Execution Time: 3.71 started 1186418640.12 ended 1186418643.83

Ended Executing this Stream at Mon Aug  6 11:44:03 2007


Stream Started at 1186418640.12
Stream Ended at 1186418643.83
Stream Processed in 3.71 seconds

----------------------------
21.sql
----------------------------
-- 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;

----------------------------
22.log
----------------------------
Begin Execution at Mon Aug  6 11:44:03 2007



-- 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 0.59 seconds.


Ended Executing this Stream at Mon Aug  6 11:44:04 2007


Stream Started at 1186418643.86
Stream Ended at 1186418644.45
Stream Processed in 0.59 seconds


SQL statements processed: 1
----------------------------
22.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:44:03 2007


Query : Execution Time: 0.59 started 1186418643.86 ended 1186418644.45

Ended Executing this Stream at Mon Aug  6 11:44:04 2007


Stream Started at 1186418643.86
Stream Ended at 1186418644.45
Stream Processed in 0.59 seconds

----------------------------
22.sql
----------------------------
-- 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;

----------------------------
3.log
----------------------------
Begin Execution at Mon Aug  6 11:43:37 2007



-- 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 0.97 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:37 2007


Stream Started at 1186418617.03
Stream Ended at 1186418617.99
Stream Processed in 0.97 seconds


SQL statements processed: 1
----------------------------
3.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:37 2007


Query : Execution Time: 0.97 started 1186418617.03 ended 1186418617.99

Ended Executing this Stream at Mon Aug  6 11:43:37 2007
```

```
Stream Started at 1186418617.03
Stream Ended at 1186418617.99
Stream Processed in 0.97 seconds

----------------------------
3.sql
----------------------------
-- 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;
----------------------------
4.log
----------------------------
Begin Execution at Mon Aug  6 11:43:38 2007



-- 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 1.14 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:39 2007


Stream Started at 1186418618.03
Stream Ended at 1186418619.16
Stream Processed in 1.14 seconds


SQL statements processed: 1
----------------------------
4.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:38 2007
```

```
Query : Execution Time: 1.14 started 1186418618.03 ended 1186418619.16

Ended Executing this Stream at Mon Aug  6 11:43:39 2007


Stream Started at 1186418618.03
Stream Ended at 1186418619.16
Stream Processed in 1.14 seconds

----------------------------
4.sql
----------------------------
-- 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;

----------------------------
5.log
----------------------------
Begin Execution at Mon Aug  6 11:43:39 2007



-- 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 1.63 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:40 2007


Stream Started at 1186418619.20
Stream Ended at 1186418620.82
Stream Processed in 1.63 seconds
```

```
SQL statements processed: 1
----------------------------
5.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:39 2007


Query : Execution Time: 1.63 started 1186418619.20 ended 1186418620.82

Ended Executing this Stream at Mon Aug  6 11:43:40 2007


Stream Started at 1186418619.20
Stream Ended at 1186418620.82
Stream Processed in 1.63 seconds

----------------------------
5.sql
----------------------------
-- 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;

----------------------------
6.log
----------------------------
Begin Execution at Mon Aug  6 11:43:40 2007



-- 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 0.13 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:40 2007


Stream Started at 1186418620.86
Stream Ended at 1186418620.99
Stream Processed in 0.13 seconds


SQL statements processed: 1
----------------------------
6.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:40 2007


Query : Execution Time: 0.13 started 1186418620.86 ended 1186418620.99
```

```
Ended Executing this Stream at Mon Aug  6 11:43:40 2007


Stream Started at 1186418620.86
Stream Ended at 1186418620.99
Stream Processed in 0.13 seconds

----------------------------
6.sql
----------------------------
-- 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;

----------------------------
7.log
----------------------------
Begin Execution at Mon Aug  6 11:43:41 2007



-- 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 0.88 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:41 2007
```

```
Stream Started at 1186418621.02
Stream Ended at 1186418621.90
Stream Processed in 0.88 seconds


SQL statements processed: 1
----------------------------
7.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:41 2007


Query : Execution Time: 0.88 started 1186418621.02 ended 1186418621.90

Ended Executing this Stream at Mon Aug  6 11:43:41 2007


Stream Started at 1186418621.02
Stream Ended at 1186418621.90
Stream Processed in 0.88 seconds

----------------------------
7.sql
----------------------------
-- 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;

----------------------------
8.log
----------------------------
Begin Execution at Mon Aug  6 11:43:41 2007



-- 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 1.20 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:43 2007


Stream Started at 1186418621.94
Stream Ended at 1186418623.14
Stream Processed in 1.20 seconds


SQL statements processed: 1
----------------------------
8.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:41 2007


Query : Execution Time: 1.20 started 1186418621.94 ended 1186418623.14

Ended Executing this Stream at Mon Aug  6 11:43:43 2007


Stream Started at 1186418621.94
Stream Ended at 1186418623.14
Stream Processed in 1.20 seconds

----------------------------
8.sql
----------------------------
-- 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;

----------------------------
9.log
----------------------------
Begin Execution at Mon Aug  6 11:43:43 2007



-- 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
ARGENTINA            1998.00        30211185.71
ARGENTINA            1997.00        50805741.75
ARGENTINA            1996.00        51923746.58
ARGENTINA            1995.00        49298625.77
ARGENTINA            1994.00        50835610.11
ARGENTINA            1993.00        51646079.18
ARGENTINA            1992.00        50410314.99
BRAZIL              1998.00         27217924.38
BRAZIL              1997.00         48378669.20


:
:
:
:
:
:
<< Lines deleted >>
:
:
:
```

```
:
UNITED STATES          1993.00          48029946.80
UNITED STATES          1992.00          48671944.50
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 3.64 seconds.


Ended Executing this Stream at Mon Aug  6 11:43:46 2007


Stream Started at 1186418623.17
Stream Ended at 1186418626.81
Stream Processed in 3.64 seconds


SQL statements processed: 1
----------------------------
9.rep
----------------------------
Begin Executing this Stream at Mon Aug  6 11:43:43 2007


Query : Execution Time: 3.64 started 1186418623.17 ended 1186418626.81

Ended Executing this Stream at Mon Aug  6 11:43:46 2007


Stream Started at 1186418623.17
Stream Ended at 1186418626.81
Stream Processed in 3.64 seconds

----------------------------
9.sql
----------------------------
-- 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;
```

Appendix E: Seed and Input Parameters

```
----------------------------
qp1.0
----------------------------
14          1996-01-01
2           44          BRASS       EUROPE
9           pale
20          misty       1997-01-01  ROMANIA
6           1993-01-01  0.09        24
17          Brand#23    JUMBO CAN
18          313
8           IRAQ        MIDDLE EAST             LARGE BRUSHED STEEL
21          VIETNAM
13          pending     accounts
3           AUTOMOBILE              1995-03-24
22          12          28          29          22          20          23
            14
16          Brand#12    MEDIUM BRUSHED          48          41          44
            39          37          38          50          22
4           1996-10-01
11          GERMANY     0.0000000333
15          1995-01-01
1           96
10          1993-02-01
19          Brand#43    Brand#54    Brand#41    10          12          26
5           ASIA        1993-01-01
7           UNITED STATES           IRAQ
12          RAIL        MAIL        1994-01-01
----------------------------
qp1.1
----------------------------
21          JORDAN
3           FURNITURE   1995-03-10
18          315
5           EUROPE      1994-01-01
11          SAUDI ARABIA            0.0000000333
7           MOZAMBIQUE              CANADA
6           1994-01-01  0.07        24
20          yellow      1995-01-01  INDONESIA
17          Brand#25    WRAP CASE
12          AIR         TRUCK       1996-01-01
16          Brand#52    PROMO BURNISHED         42          50          31
            19          41          44          26          14
15          1997-01-01
13          pending     accounts
10          1993-11-01
2           31          NICKEL      AMERICA
8           CANADA      AMERICA     LARGE PLATED STEEL
14          1996-01-01
19          Brand#55    Brand#32    Brand#45    6           13          22
9           moccasin
22          19          30          31          22          16          27
            17
1           104
4           1994-07-01
----------------------------
qp1.2
----------------------------
6           1994-01-01  0.04        25
17          Brand#22    WRAP JAR
14          1996-01-01
16          Brand#32    SMALL PLATED            10          8           49
            5           30          2           12          38
19          Brand#53    Brand#25    Brand#34    1           14          29
10          1994-09-01
9           maroon
2           19          TIN         MIDDLE EAST
15          1995-01-01
8           SAUDI ARABIA            MIDDLE EAST             LARGE ANODIZED
COPPER
5           MIDDLE EAST             1994-01-01
22          23          14          27          24          20          22
            33
12          REG AIR     TRUCK       1996-01-01
7           INDIA       SAUDI ARABIA
13          pending     accounts
18          313
1           112
4           1997-02-01
20          ivory       1994-01-01  UNITED KINGDOM
3           AUTOMOBILE              1995-03-26
11          INDIA       0.0000000333
21          ETHIOPIA
----------------------------
qp1.3
----------------------------
8           JAPAN       ASIA        MEDIUM POLISHED COPPER
5           AFRICA      1994-01-01
4           1994-11-01
6           1994-01-01  0.02        25
17          Brand#34    WRAP CAN
7           ALGERIA     JAPAN
1           120
18          314
22          17          31          10          24          13          12
            28
14          1996-01-01
9           lawn
10          1993-06-01
15          1997-01-01
11          VIETNAM     0.0000000333
20          seashell    1997-01-01  JORDAN
2           7           COPPER      AMERICA
21          UNITED KINGDOM
19          Brand#55    Brand#53    Brand#33    6           15          25
13          unusual     deposits
16          Brand#13    LARGE BRUSHED           7           11          39
            28          26          4           29          15
12          SHIP        TRUCK       1996-01-01
3           FURNITURE   1995-03-12
----------------------------
qp1.4
----------------------------
5           AMERICA     1994-01-01
21          MOROCCO
14          1997-01-01
19          Brand#12    Brand#41    Brand#33    2           16          21
15          1995-01-01
17          Brand#31    SM CASE
12          FOB         TRUCK       1997-01-01
6           1994-01-01  0.07        24
4           1997-06-01
9           hot
8           EGYPT       MIDDLE EAST             MEDIUM BURNISHED COPPER
16          Brand#53    STANDARD ANODIZED       10          13          31
            39          23          19          22          28
11          INDONESIA   0.0000000333
2           45          BRASS       MIDDLE EAST
10          1994-03-01
18          312
1           67
13          unusual     deposits
7           PERU        EGYPT
22          13          21          10          17          15          19
            18
3           MACHINERY               1995-03-28
20          deep        1996-01-01  CANADA
----------------------------
qp1.5
----------------------------
21          GERMANY
15          1993-01-01
4           1995-03-01
6           1994-01-01  0.04        25
7           INDONESIA   VIETNAM
16          Brand#33    MEDIUM PLATED           5           34          28
            37          19          13          32          36
19          Brand#14    Brand#24    Brand#22    7           17          29
18          313
14          1997-01-01
22          28          34          16          25          18          30
            12
11          RUSSIA      0.0000000333
13          unusual     deposits
3           BUILDING    1995-03-14
1           75
2           32          NICKEL      ASIA
5           ASIA        1994-01-01
8           VIETNAM     ASIA        SMALL BRUSHED COPPER
20          papaya      1994-01-01  CHINA
12          MAIL        TRUCK       1997-01-01
17          Brand#33    SM JAR
10          1994-12-01
9           gainsboro
----------------------------
qp1.6
----------------------------
10          1993-09-01
3           MACHINERY               1995-03-30
15          1995-01-01
13          unusual     deposits
6           1995-01-01  0.02        25
8           JORDAN      MIDDLE EAST             SMALL POLISHED TIN
9           dodger
7           ARGENTINA JORDAN
```

```
4           1997-10-01
11          IRAN        0.0000000333
22          27          25          24          11          15          32
            26
18          315
12          RAIL        MAIL        1997-01-01
1           83
5           EUROPE      1995-01-01
16          Brand#13    ECONOMY POLISHED        27          21          19
            16          18          45          22          46
2           20          TIN         MIDDLE EAST
14          1997-01-01
19          Brand#11    Brand#52    Brand#21    2           18          25
20          blanched    1997-01-01  GERMANY
17          Brand#34    SM CAN
21          UNITED STATES
----------------------------
qp1.7
----------------------------
18          312
8           ETHIOPIA    AFRICA      SMALL BURNISHED TIN
20          linen       1996-01-01  RUSSIA
21          MOZAMBIQUE
2           8           COPPER      ASIA
4           1995-06-01
22          32          24          16          11          30          27
            21
17          Brand#31    LG CASE
1           91
11          UNITED KINGDOM          0.0000000333
9           cornsilk
19          Brand#23    Brand#45    Brand#25    7           19          21
3           BUILDING    1995-03-16
13          unusual     deposits
5           MIDDLE EAST             1995-01-01
7           CHINA       ETHIOPIA
10          1994-07-01
16          Brand#53    SMALL ANODIZED          12          26          34
            31          30          48          9           15
6           1995-01-01  0.07        24
14          1998-01-01
15          1993-01-01
12          AIR         MAIL        1997-01-01
----------------------------
qp1.8
----------------------------
19          Brand#25    Brand#23    Brand#14    3           20          28
1           99
15          1996-01-01
17          Brand#33    LG JAR
5           AFRICA      1995-01-01
8           RUSSIA      EUROPE      STANDARD BRUSHED TIN
9           burnished
12          REG AIR     MAIL        1993-01-01
14          1993-01-01
7           IRAN        RUSSIA
4           1993-03-01
3           HOUSEHOLD               1995-03-01
20          tan         1994-01-01  JAPAN
16          Brand#33    LARGE BURNISHED         15          43          18
            16          9           33          28          5
6           1995-01-01  0.05        25
22          22          14          26          28          32          12
            10
10          1993-04-01
13          unusual     packages
2           46          STEEL       AFRICA
21          INDIA
18          314
11          IRAQ        0.0000000333
----------------------------
qp1.9
----------------------------
8           KENYA       AFRICA      STANDARD PLATED TIN
13          unusual     packages
2           34          NICKEL      ASIA
20          ghost       1993-01-01  BRAZIL
17          Brand#35    LG CAN
3           BUILDING    1995-03-18
6           1995-01-01  0.02        25
21          ARGENTINA
18          315
11          UNITED STATES           0.0000000333
19          Brand#22    Brand#11    Brand#14    8           10          24
10          1994-01-01
15          1993-01-01
4           1995-10-01
22          14          23          25          27          32          17
            33
1           107
7           BRAZIL      KENYA
12          SHIP        MAIL        1993-01-01
9           black
```

```
14          1993-01-01
5           AMERICA     1995-01-01
16          Brand#13    PROMO POLISHED          8           17          40
            24          4           46          21          18
----------------------------
qp2.0
----------------------------
14          1996-01-01
2           44          BRASS       EUROPE
9           pale
20          misty       1997-01-01  ROMANIA
6           1993-01-01  0.09        24
17          Brand#23    JUMBO CAN
18          313
8           IRAQ        MIDDLE EAST             LARGE BRUSHED STEEL
21          VIETNAM
13          pending     accounts
3           AUTOMOBILE              1995-03-24
22          12          28          29          22          20          23
            14
16          Brand#12    MEDIUM BRUSHED          48          41          44
            39          37          38          50          22
4           1996-10-01
11          GERMANY     0.0000000333
15          1995-01-01
1           96
10          1993-02-01
19          Brand#43    Brand#54    Brand#41    10          12          26
5           ASIA        1993-01-01
7           UNITED STATES           IRAQ
12          RAIL        MAIL        1994-01-01
----------------------------
qp2.1
----------------------------
21          JORDAN
3           FURNITURE   1995-03-10
18          315
5           EUROPE      1994-01-01
11          SAUDI ARABIA            0.0000000333
7           MOZAMBIQUE              CANADA
6           1994-01-01  0.07        24
20          yellow      1995-01-01  INDONESIA
17          Brand#25    WRAP CASE
12          AIR         TRUCK       1996-01-01
16          Brand#52    PROMO BURNISHED         42          50          31
            19          41          44          26          14
15          1997-01-01
13          pending     accounts
10          1993-11-01
2           31          NICKEL      AMERICA
8           CANADA      AMERICA     LARGE PLATED STEEL
14          1996-01-01
19          Brand#55    Brand#32    Brand#45    6           13          22
9           moccasin
22          19          30          31          22          16          27
            17
1           104
4           1994-07-01
----------------------------
qp2.2
----------------------------
6           1994-01-01  0.04        25
17          Brand#22    WRAP JAR
14          1996-01-01
16          Brand#32    SMALL PLATED            10          8           49
            5           30          2           12          38
19          Brand#53    Brand#25    Brand#34    1           14          29
10          1994-09-01
9           maroon
2           19          TIN         MIDDLE EAST
15          1995-01-01
8           SAUDI ARABIA            MIDDLE EAST             LARGE ANODIZED
COPPER
5           MIDDLE EAST             1994-01-01
22          23          14          27          24          20          22
            33
12          REG AIR     TRUCK       1996-01-01
7           INDIA       SAUDI ARABIA
13          pending     accounts
18          313
1           112
4           1997-02-01
20          ivory       1994-01-01  UNITED KINGDOM
3           AUTOMOBILE              1995-03-26
11          INDIA       0.0000000333
21          ETHIOPIA
----------------------------
qp2.3
----------------------------
8           JAPAN       ASIA        MEDIUM POLISHED COPPER
5           AFRICA      1994-01-01
4           1994-11-01
6           1994-01-01  0.02        25
```

```
17        Brand#34  WRAP CAN
7         ALGERIA   JAPAN
1         120
18        314
22        17        31        10        24        13        12
          28
14        1996-01-01
9         lawn
10        1993-06-01
15        1997-01-01
11        VIETNAM   0.0000000333
20        seashell  1997-01-01 JORDAN
2         7         COPPER    AMERICA
21        UNITED KINGDOM
19        Brand#55  Brand#53  Brand#33  6         15        25
13        unusual   deposits
16        Brand#13  LARGE BRUSHED       7         11        39
          28        26        4         29        15
12        SHIP      TRUCK     1996-01-01
3         FURNITURE 1995-03-12
----------------------------
qp2.4
----------------------------
5         AMERICA   1994-01-01
21        MOROCCO
14        1997-01-01
19        Brand#12  Brand#41  Brand#33  2         16        21
15        1995-01-01
17        Brand#31  SM CASE
12        FOB       TRUCK     1997-01-01
6         1994-01-01 0.07     24
4         1997-06-01
9         hot
8         EGYPT     MIDDLE EAST         MEDIUM BURNISHED COPPER
16        Brand#53  STANDARD ANODIZED   10        13        31
          39        23        19        22        28
11        INDONESIA 0.0000000333
2         45        BRASS     MIDDLE EAST
10        1994-03-01
18        312
1         67
13        unusual   deposits
7         PERU      EGYPT
22        13        21        10        17        15        19
          18
3         MACHINERY           1995-03-28
20        deep      1996-01-01 CANADA
----------------------------
qp2.5
----------------------------
21        GERMANY
15        1993-01-01
4         1995-03-01
6         1994-01-01 0.04     25
7         INDONESIA VIETNAM
16        Brand#33  MEDIUM PLATED       5         34        28
          37        19        13        32        36
19        Brand#14  Brand#24  Brand#22  7         17        29
18        313
14        1997-01-01
22        28        34        16        25        18        30
          12
11        RUSSIA    0.0000000333
13        unusual   deposits
3         BUILDING  1995-03-14
1         75
2         32        NICKEL    ASIA
5         ASIA      1994-01-01
8         VIETNAM   ASIA      SMALL BRUSHED COPPER
20        papaya    1994-01-01 CHINA
12        MAIL      TRUCK     1997-01-01
17        Brand#33  SM JAR
10        1994-12-01
9         gainsboro
----------------------------
qp2.6
----------------------------
10        1993-09-01
3         MACHINERY           1995-03-30
15        1995-01-01
13        unusual   deposits
6         1995-01-01 0.02     25
8         JORDAN    MIDDLE EAST         SMALL POLISHED TIN
9         dodger
7         ARGENTINA JORDAN
4         1997-10-01
11        IRAN      0.0000000333
22        27        25        24        11        15        32
          26
18        315
12        RAIL      MAIL      1997-01-01
1         83
5         EUROPE    1995-01-01
16        Brand#13  ECONOMY POLISHED    27        21        19
          16        18        45        22        46
2         20        TIN       MIDDLE EAST
14        1997-01-01
19        Brand#11  Brand#52  Brand#21  2         18        25
20        blanched  1997-01-01 GERMANY
17        Brand#34  SM CAN
21        UNITED STATES
----------------------------
qp2.7
----------------------------
18        312
8         ETHIOPIA  AFRICA    SMALL BURNISHED TIN
20        linen     1996-01-01 RUSSIA
21        MOZAMBIQUE
2         8         COPPER    ASIA
4         1995-06-01
22        32        24        16        11        30        27
          21
17        Brand#31  LG CASE
1         91
11        UNITED KINGDOM      0.0000000333
9         cornsilk
19        Brand#23  Brand#45  Brand#25  7         19        21
3         BUILDING  1995-03-16
13        unusual   deposits
5         MIDDLE EAST         1995-01-01
7         CHINA     ETHIOPIA
10        1994-07-01
16        Brand#53  SMALL ANODIZED      12        26        34
          31        30        48        9         15
6         1995-01-01 0.07     24
14        1998-01-01
15        1993-01-01
12        AIR       MAIL      1997-01-01
----------------------------
qp2.8
----------------------------
19        Brand#25  Brand#23  Brand#14  3         20        28
1         99
15        1996-01-01
17        Brand#33  LG JAR
5         AFRICA    1995-01-01
8         RUSSIA    EUROPE    STANDARD BRUSHED TIN
9         burnished
12        REG AIR   MAIL      1993-01-01
14        1993-01-01
7         IRAN      RUSSIA
4         1993-03-01
3         HOUSEHOLD           1995-03-01
20        tan       1994-01-01 JAPAN
16        Brand#33  LARGE BURNISHED     15        43        18
          16        9         33        28        5
6         1995-01-01 0.05     25
22        22        14        26        28        32        12
          10
10        1993-04-01
13        unusual   packages
2         46        STEEL     AFRICA
21        INDIA
18        314
11        IRAQ      0.0000000333
----------------------------
qp2.9
----------------------------
8         KENYA     AFRICA    STANDARD PLATED TIN
13        unusual   packages
2         34        NICKEL    ASIA
20        ghost     1993-01-01 BRAZIL
17        Brand#35  LG CAN
3         BUILDING  1995-03-18
6         1995-01-01 0.02     25
21        ARGENTINA
18        315
11        UNITED STATES       0.0000000333
19        Brand#22  Brand#11  Brand#14  8         10        24
10        1994-01-01
15        1993-01-01
4         1995-10-01
22        14        23        25        27        32        17
          33
1         107
7         BRAZIL    KENYA
12        SHIP      MAIL      1993-01-01
9         black
14        1993-01-01
5         AMERICA   1995-01-01
16        Brand#13  PROMO POLISHED      8         17        40
          24        4         46        21        18
----------------------------
seed
----------------------------
803160408
```

Appendix F: Benchmark Scripts

```
----------------------------
runTPCHbuild
----------------------------
#!/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`
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

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.$RUN_ID
touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
#/home/oracle/kit/schema/10.0/build/dbcre_10gR2_cclass.sh >> $LD1DBCRE
#/home/oracle/kit/schema/10.0/build/tscre_10gR2.sh >>  $LD2SCTSO
#~/frame/bin/2start
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
/home/oracle/kit/schema/10.0/build/dapop_10gR2_mp2.sh >> $LD3DAPOP
/home/oracle/kit/schema/10.0/build/create_et2_driver.sh >> $LD3DAPOP
$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo Generated seed: `cat $KIT_DIR/audit/seed` >> $SCRIPT_LOG_FILE
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
----------------------------
runTPCHpt
----------------------------
#!/bin/ksh
. $KIT_DIR/env
#ECHO=/bin/echo
SCRIPT_DIR=${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

${ECHO} ${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
${ECHO} ${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

#sleep 400
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
----------------------------
runTPCHrun
----------------------------
#!/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

RUN_ID=`cat $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

ckpnt.sh
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
runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}
ckpnt.sh
ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}


cp $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log $OUT_DIR

echo "End TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID `date`" >>
$SCRIPT_LOG_FILE
----------------------------
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
----------------------------

#
# $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=64

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`

sqlplus /NOLOG << !

connect $PASSWD;
set timing on
set serveroutput on
set echo on

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${SETNUM});

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${SETNUM});

commit;
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=128
PAR_HINT1=4
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

create table temp_okey${SETNUM} tablespace TSD1 parallel ${PAR_HINT1} nologging as
select * from temp_okey_et${SETNUM};

create unique index i_temp_okey${SETNUM} on temp_okey${SETNUM} (t_orderkey)
tablespace TSD1 parallel ${PAR_HINT1}  nologging compute statistics;

analyze table temp_okey${SETNUM} estimate statistics sample 1 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${SETNUM} t where
o.o_orderkey = t.t_orderkey order by 1);

delete from (select /*+ use_nl(l) */ l.rowid from lineitem l,temp_okey${SETNUM} t where
l.l_orderkey = t.t_orderkey order by 1);

commit;
exit;
!

END=`$GTIME`

# Done

echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
```

Appendix G: Price Quotes

From: MaryBeth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Monday, August 06, 2007 7:36 PM
To: Nambiar, Raghu Othayoth
Subject: Oracle Pricing

Product	**Price**	**Quantity**	**Extended Price**
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 years	10,000	16*	160,000
Oracle Real Application Clusters, Named User Please for 3 years	5,000	16*	80,000
Partitioning, Named User Plus for 3 years	2,500	16*	40,000
Oracle Database Server Support Package for 3 years	16,000		48,000
Oracle Mandatory E-Business Discount			<65,600>
Total			262,400

* 16 = 0.50 * 32. Explanation: For the purposes of counting the number of processors which require licensing, an Intel multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of .50.
Oracle Pricing Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081.