PANTA Systems

TPC Benchmark™ H
Full Disclosure Report

PANTA Systems PANTAmatrix
using
Oracle Database 10g Enterprise Edition Release 2 with
Oracle Real Application Clusters and Partitioning and
Red Hat Enterprise Linux 4 Advanced Server Update 3

First Edition
October 2006

PANTA Systems PANTAmatrix using Oracle Database 10g Enterprise Edition Release 2
with Oracle Real Application Clusters and Partitioning
and Red Hat Enterprise Linux 4 Advanced Server Update 3

First Edition – October 2006

PANTA Systems Inc., the sponsor of this benchmark test, believes that the technical, pricing, and discounting information contained in this document is accurate as of its publication date. Such information within the document is subject to change without notice.

All performance data contained in this report was obtained in a controlled lab setting. Results obtained in other environments may vary widely. This report does not guarantee or imply any warranty of system performance, nor will PANTA Systems assume responsibility for any errors that may appear.

Benchmark results are highly dependent on the exact combination of hardware and software components, configurations, and settings. Therefore, customers should not use TPC-H benchmark results in lieu of specific application benchmarks for critical capacity planning or product evaluation decisions.

PANTAmatrix is a registered trademark of PANTA Systems, Inc.

Oracle Database 10G is a registered trademark of Oracle Corporation. Red Hat Enterprise Linux is a registered trademark of Red Hat Inc. AMD and AMD Opteron are trademarks of Advanced Micro Devices, Inc.

TPC Benchmark H is a trademark of the Transaction Processing Performance Council.

All other brand or product names mentioned are considered trademarks or registered trademarks of their respective owners.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark H test conducted on the PANTA Systems PANTAmatrix using Oracle Database 10g Enterprise Edition Release 2 with Oracle Real Application Clusters and Partitioning in conformance with the requirements of the TPC-H Benchmark Specification. The operating system used for the benchmark was Red Hat Enterprise Linux 4 Advanced Server Update 3.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	QphH @1000GB	$ / QphH @1000GB
PANTA Systems PANTAmatrix 64P PANTA Systems Storage Array Modules	Oracle Database 10g Enterprise Edition Release 2 with Oracle Real Application Clusters and Partitioning Red Hat Enterprise Linux 4 AS Update 3	$1,480,369.65	59,353.9	$24.94

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

Standard and Executive Summary Statements

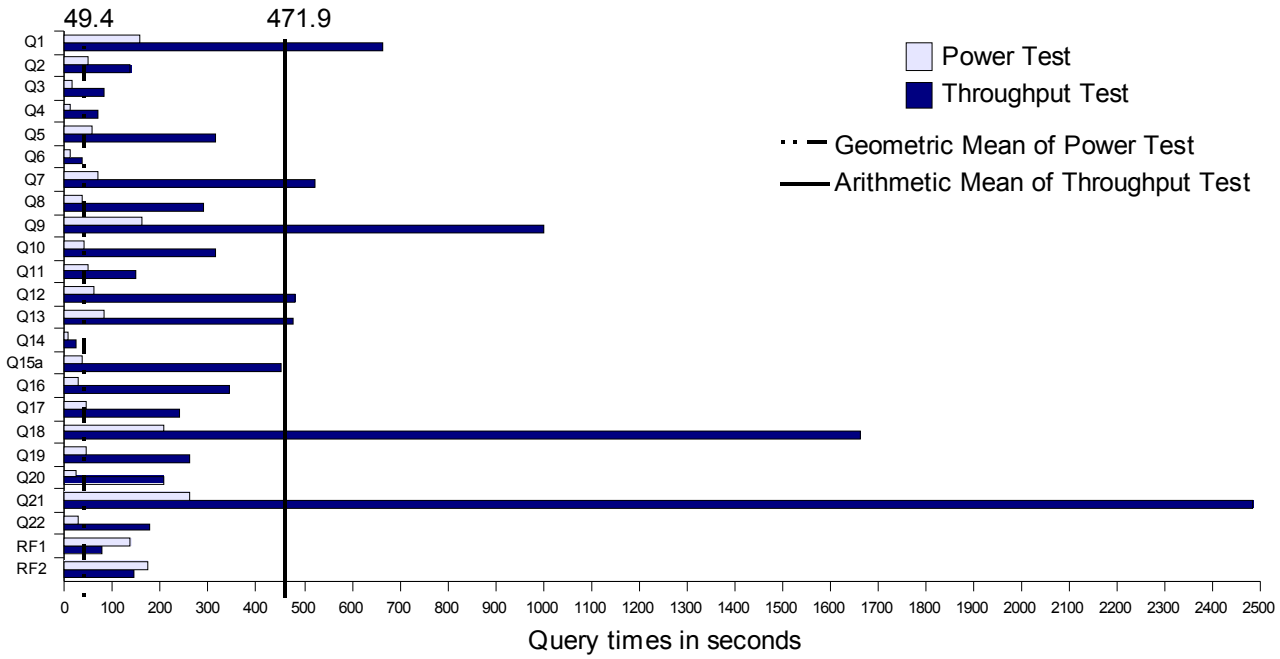
The Executive Summary and Numerical Quantities Summary of the benchmark results for the PANTA Systems PANTAmatrix can be found in the following pages.

Auditor

In order to verify compliance to the TPC-H benchmark specification, Francois Raab, Infosizing, Inc., audited the benchmark configuration, environment, and methodology used to produce and validate the test results, and the pricing model used to calculate the price/performance.

PANTA Reshaping the Server™	PANTA Systems PANTAmatrix	TPC-H Rev. 2.5.0 Report Date: **October 19, 2006**
Total System Cost	Composite Query per Hour Metric	Price / Performance
$1,480,369.65	**59,353.9** QphH@1000GB	**$24.94** $ / QphH@1000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
1000 GB	**Oracle Database 10g Enterprise Edition Release 2 with Oracle Real Application Clusters and Partitioning**	**Red Hat Enterprise Linux 4 Advanced Server Update 3**	**None**	**April 15, 2007**



Database Load Time = 6:07:59	Load Included Backup: Y	Total Data Storage / Database Size = 133.0
RAID (Base tables only): N	RAID (Base tables and auxiliary data structures): N	RAID (All): N

System Components	System Total	Per Node
Nodes:	8	n.a.
Processor (Dual-Core AMD Opteron™ processor Model 875HE 2.2GHz/1MB L2 cache per core for 2MB total/1GHz Hyper Transport:	32	4
Cores:	64	8
Threads:	64	8
Memory:	256GB	32GB
OS Storage (PANTA Systems PANTAmatrix System Storage Module):	1	n.a.
OS Disk Drives (250GB 7200RPM SATA HDD):	14	n.a.
Network Interfaces (Infiniband 4x ports):	32	4
Storage Switch (PANTA Systems 24-port Infiniband switch):	2	n.a.
Storage Switch (Silverstorm Technologies 9024 24-port Infiniband switch):	2	n.a.
Storage Subsystem (PANTA Systems Storage Array Module):	37	n.a.
Storage Subsystem Disk Drives (250GB 7200RPM SATA HDD):	518	n.a.
Total Storage:	133,000GB	n.a.

PANTA™

Reshaping the Server™

PANTA Systems PANTAmatrix

TPC-H Rev. 2.5.0

Report Date:
October 19, 2006

Description	Part Number	Brand	Pricing	Unit Price	Qty	Extended Price	3 yr. Maint. Price
System Hardware							
Compute Module; 4 x 2.2GHz DUAL CORE (Opteron 875HE) processors, NO memory, 16 available DIMM slots, dual port InfiniBand 4X, single PCIe 8X slot	CPE-0424		1	26,000.00	8	208,000.00	80,500.00
16GB Memory Kit (Kit contains 8x2GB DDR-400MHz ECC Registered DIMMs), for use with 4-socket Compute Module	MEM-0116		1	6,100.00	16	97,600.00	
PANTA 8U Enclosure for Compute, Visualization, and Switch Modules	SUB-0100		1	5,400.00	1	5,400.00	
PANTA Switch Module, 24 port based InfiniBand switch with 8 connections to Compute Modules and 16 externally accessible connections	SWE-0124-A		1	6,300.00	2	12,600.00	
Silverstorm 9024 (24-port InfiniBand SDR switch w/ embedded management support, sigle power supply and subnet manager key for up to 288 nodes).	SWE-S124		1	10,100.00	2	20,200.00	4,600.00
PANTA 42U rack with standard front door	RCK-0100		1	6,700.00	3	20,100.00	
Rack side panels	RCK-0101		1	0.00	1	0.00	
42U rack rear door - standard	RCK-0102		1	0.00	3	0.00	
PANTA 42U Tall Rack Shipping Crate	RCK-0900		1	0.00	3	0.00	
3 x powercables from PANTA 8U enclosure to rack based power distribution unit	CBL-0001		1	80.00	1	80.00	
16 port Gigabit Ethernet switch for PANTA management network	HUB-0001		1	550.00	3	1,650.00	
PANTA Rack mounted power distribution units (includes 2 PDU's for full rack configuration, total of 4 cords, 220V 3Phase-20A)	PWR-0001		1	3,800.00	3	11,400.00	
3M InfiniBand Cable	CBL-0103		1	105.00	19	1,995.00	
Ethernet Cable (RJ45 Cat-5E) 7 feet	CBL-0041		1	5.25	4	21.00	
2 port InfiniBand 4X HBA (PCIe based)	IOE-0002		1	1,500.00	8	12,000.00	
					Subtotal	**391,046.00**	**85,100.00**
Storage and System Management Hardware							
PANTAmatrix System Control Module with dual port InfiniBand 4X HBA, and PANTA Sys-Manager software	SCM-0210		1	6,800.00	1	6,800.00	3,840.00
PANTAmatrix System Storage Module with dual port InfiniBand 4X support, dual RAID controllers with battery backup, 14 externally accessible drive bays (empty), battery backup, and PANTA Sys-Manager Software	SSM-0310		1	13,400.00	1	13,400.00	
PANTA Storage Array Module with dual port InfiniBand 4X support, dual RAID controllers with battery backup, and 14 externally accessible drive bays (empty).	SAM-0510		1	13,400.00	37	495,800.00	149,500.00
Sled mounted 250GB 7200 RPM SATA drive for the SCM-0210, SCE-0300, SSM-0310, and SAM-0510	HD3-250GB		1	280.00	532	148,960.00	
3M InfiniBand Cable	CBL-0103		1	105.00	39	4,095.00	
Ethernet Cable (RJ45 Cat-5E) 7 feet	CBL-0041		1	5.25	39	204.75	
					Subtotal	**669,259.75**	**153,340.00**
Software							
PANTA Sys-Manager 2.0 License for up to 8 Compute Modules	PSM-0020		1	1,500.00	1	1,500.00	1,500.00
PANTA Sys-Manager Media Kit	PSM-0001-MK		1	150.00	1	150.00	
Red Hat Enterprise Linux AS 4.0, Update 3, Standard Edition (single node license, 1yr support)	RHAS-0001		1	2,500.00	8	20,000.00	40,000.00
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 Years	run-time	Oracle	2	10,000.00	32	320,000.00	
Oracle Real Appliaction Clusters, Named User Plus for 3 Years	run-time	Oracle	2	5,000.00	32	160,000.00	
Partitioning, Named User Plus for 3 Years	run-time	Oracle	2	2,500.00	32	80,000.00	
Database Server Support Package for 3 Years	run-time	Oracle	2	16,000.00	3		48,000.00
					Subtotal	**581,650.00**	**89,500.00**
PANTA Large Purchase and Net30 Discount			1			-367,926.10	
Oracle Mandatory E-Business Discount			2			-121,600.00	
					Total	**1,152,429.65**	**327,940.00**

Pricing: 1- PANTA Systems Direct
Pricing: 2- Oracle at 15% discount, Oracle pricing contact: Mary Beth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081.
**32 = 0.50 * 64. Explanation: For the purposes of counting the number of processors which require licensing, an AMD multicore chip with "n" cores shall be determined by multiplying "n" cores by a factor of 0.50.
All discounts are based on US list prices and for similar quantities and configurations.

3-Year Cost of Ownership: $1,480,369.65
QphH Rating: 59,353.9
$/QphH@1000GB: $24.94

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

Numerical Quantities

Measurement Results

Database Scale Factor	= 1,000
Total Data Storage / Database Size	= 133.0
Start of Database Load	= 10/18/2006 18:13:18
End of Database Load	= 10/19/2006 00:05:30
Database Load Time	= 6:07:59 *
Query Streams for Throughput Test	= 8
TPC-H Power	= 72,848.5
TPC-H Throughput	= 48,359.0
TPC-H Composite Query-per-Hour Metric (QphH@1000GB)	= 59,353.9
Total System Price Over 3 Years	= $1,480,369.65
TPC-H Price/ Performance Metric ($/QphH@300GB)	= $24.94

Measurement Intervals

Measurement Interval in Throughput Test	= 13,102 seconds

Duration of Stream Execution

Power Stream	Seed	RF1 Start Time RF1 End Time	Query Start Time Query End Time	RF2 Start Time RF2 End Time	Duration
	1019000530	10/19/06 07:21:47 10/19/06 07:24:02	10/19/06 07:24:02 10/19/06 07:48:39	10/19/06 07:48:39 10/19/06 07:51:33	0:29:46

Throughput Stream	Seed	Query Start Time Query End Time	RF1 Start Time RF1 End Time	RF2 Start Time RF2 End Time	Duration
1	1019000531	10/19/06 07:51:35 10/19/06 10:53:17	10/19/06 11:00:25 10/19/06 11:01:57	10/19/06 11:01:57 10/19/06 11:04:33	3:12:58
2	1019000532	10/19/06 07:51:35 10/19/06 10:54:26	10/19/06 11:04:33 10/19/06 11:05:59	10/19/06 11:05:59 10/19/06 11:08:30	3:16:55
3	1019000533	10/19/06 07:51:35 10/19/06 10:38:03	10/19/06 11:08:30 10/19/06 11:09:53	10/19/06 11:09:53 10/19/06 11:12:21	3:20:46
4	1019000534	10/19/06 07:51:35 10/19/06 10:30:23	10/19/06 11:12:21 10/19/06 11:13:41	10/19/06 11:13:41 10/19/06 11:16:09	3:24:34
5	1019000535	10/19/06 07:51:35 10/19/06 11:00:24	10/19/06 11:16:09 10/19/06 11:17:22	10/19/06 11:17:22 10/19/06 11:19:43	3:28:08
6	1019000536	10/19/06 07:51:35 10/19/06 10:57:23	10/19/06 11:19:43 10/19/06 11:20:58	10/19/06 11:20:58 10/19/06 11:23:24	3:31:49
7	1019000537	10/19/06 07:51:35 10/19/06 10:50:30	10/19/06 11:23:24 10/19/06 11:24:33	10/19/06 11:24:33 10/19/06 11:26:43	3:35:08
8	1019000538	10/19/06 07:51:35 10/19/06 10:12:30	10/19/06 11:26:43 10/19/06 11:27:48	10/19/06 11:27:48 10/19/06 11:29:57	3:38:22

* A backup of the database partitions is required to meet the durability requirements. The backup time (15 min. 47 sec.) was added into the elapsed load time.

TPC-H Timing Intervals (in seconds)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Stream 0	155.5	47.7	16.7	10.4	54.7	10.4	67.4	33.9	161.0	41.3	50.3	59.5
Stream 1	216.7	151.3	126.6	24.1	283.6	31.4	852.4	203.1	543.8	465.3	122.8	536.5
Stream 2	962.2	159.8	135.0	95.9	213.2	21.2	529.9	135.8	874.1	298.5	170.6	649.0
Stream 3	739.5	119.3	71.1	45.3	211.4	32.1	637.0	255.9	755.7	399.4	145.1	275.2
Stream 4	619.3	121.4	49.9	119.7	253.4	44.1	249.1	325.4	2341.8	484.7	129.9	649.2
Stream 5	234.3	107.2	36.3	72.6	378.4	31.5	647.1	139.3	271.3	65.9	132.8	330.0
Stream 6	838.2	105.4	89.1	26.4	355.8	78.0	228.5	402.4	1814.2	229.0	108.8	743.9
Stream 7	801.3	223.5	45.4	114.2	198.3	29.5	718.0	609.6	716.8	152.6	234.0	314.7
Stream 8	907.8	83.6	88.3	52.3	617.1	30.6	311.7	242.8	687.8	418.8	127.0	346.8
Minimum	216.7	83.6	36.3	24.1	198.3	21.2	228.5	135.8	271.3	65.9	108.8	275.2
Average	664.9	133.9	80.2	68.8	313.9	37.3	521.7	289.3	1000.7	314.3	146.3	480.7
Maximum	962.2	223.5	135.0	119.7	617.1	78.0	852.4	609.6	2341.8	484.7	234.0	743.9

	Q13	Q14	Q15a	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 0	81.1	8.4	37.9	28.9	44.4	208.3	45.4	24.2	260.8	29.2	135.5	173.3
Stream 1	649.6	24.9	924.7	446.9	227.1	1538.9	156.9	310.1	2961.4	102.0	92.9	155.9
Stream 2	361.1	25.4	128.6	176.0	539.0	1863.1	175.8	239.6	3047.0	169.6	85.2	151.5
Stream 3	597.1	20.4	153.1	623.6	184.8	624.6	221.5	143.9	3577.5	153.6	83.2	147.4
Stream 4	339.7	20.3	243.8	159.8	245.7	1379.7	243.4	92.1	1279.8	133.2	80.2	148.2
Stream 5	211.9	27.6	170.0	307.3	110.4	4059.5	681.8	112.5	2903.0	298.3	72.6	141.5
Stream 6	1085.0	17.1	854.6	829.8	175.1	818.6	136.9	111.3	1991.9	107.5	74.3	146.6
Stream 7	320.4	12.9	212.6	98.6	222.1	1988.4	170.0	329.2	2885.5	336.5	68.4	130.3
Stream 8	260.4	21.9	920.4	120.0	218.0	1031.3	304.8	306.6	1235.6	121.0	65.5	129.0
Minimum	211.9	12.9	128.6	98.6	110.4	624.6	136.9	92.1	1235.6	102.0	65.5	129.0
Average	478.1	21.3	451.0	345.3	240.3	1663.0	261.4	205.7	2485.2	177.7	77.8	143.8
Maximum	1085.0	27.6	924.7	829.8	539.0	4059.5	681.8	329.2	3577.5	336.5	92.9	155.9

Table of Contents

1.0 General Items

1.1 Test Sponsor

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

PANTA Systems Inc. sponsored this benchmark. The benchmark was developed and engineered by PANTA Systems Inc. and Oracle Corporation.

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 but 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.

Details of system and database configurations and parameters are provided in Appendix A.

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 PANTA Systems PANTAmatrix 64P (depicted in Figure 1.1) that was used to obtain the results in this benchmark consists of:

System Components	System Total	Per Node
Nodes:	8	n.a.
Processor (Dual-Core AMD Opteron™ processor Model 875HE 2.2GHz/1MB L2 cache per core for 2MB total/1GHz Hyper Transport:	32	4
Cores:	64	8
Threads:	64	8
Memory:	256GB	32GB
OS Storage (PANTA Systems PANTAmatrix System Storage Module):	1	n.a.
OS Disk Drives (250GB 7200RPM SATA HDD):	14	n.a.
Network Interfaces (Infiniband 4x ports):	32	4
Storage Switch (PANTA Systems 24-port Infiniband switch):	2	n.a.
Storage Switch (Silverstorm Technologies 9024 24-port Infiniband switch):	2	n.a.
Storage Subsystem (PANTA Systems Storage Array Module):	37	n.a.
Storage Subsystem Disk Drives (250GB 7200RPM SATA HDD):	518	n.a.
Total Storage:	133,000GB	n.a.

Each of the eight nodes are equipped with two Infiniband HCAs, each with two Infiniband 4x ports. The two ports of one HCA connect to separate PANTA Systems 24-port Infiniband switches. The two ports of the second HCA connect to separate Silverstorm Technologies 9024 24-port Infiniband switches.

The storage subsystem consists of 37 PANTA Systems Storage Array Modules. Disk arrays 1-8 connect to the first PANTA Systems 24-port Infiniband switch. Disk arrays 9-17 connect to the second PANTA Systems 24-port Infiniband switch. Disk arrays 18-27 connect to the first Silverstorm Technologies 24-port Infiniband switch. Disk Arrays 28-37 connect to the second Silverstorm Technologies 24-port Infiniband switch.

The PANTAmatrix System Storage Module, used to centrally store the root filesystems (boot disks) and swap spaces for the 8 nodes, is connected to the first PANTA Systems 24-port Infiniband switch.

Each PANTA Systems Storage Array Module contains 14 250GB 7200RPM SATA hard drives. The disks in 34 of the arrays are configured in sets of 2-disk hardware RAID0. Therefore each of those arrays contain 7 RAID0 sets containing 2 disks each. The remaining 3 arrays contain 6 RAID0 sets containing 2 disks each. The additional 2 drives in those arrays are configured as individual passthru

disks.

All disks are directly accessible from all nodes via Infiniband SRP. The 256 RAID0 disks are used for Oracle tables, indexes, temp tablespace, redo/undo log files, and flatfile storage. The 6 remaining individual disks are used as Oracle config and lock partitions.

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

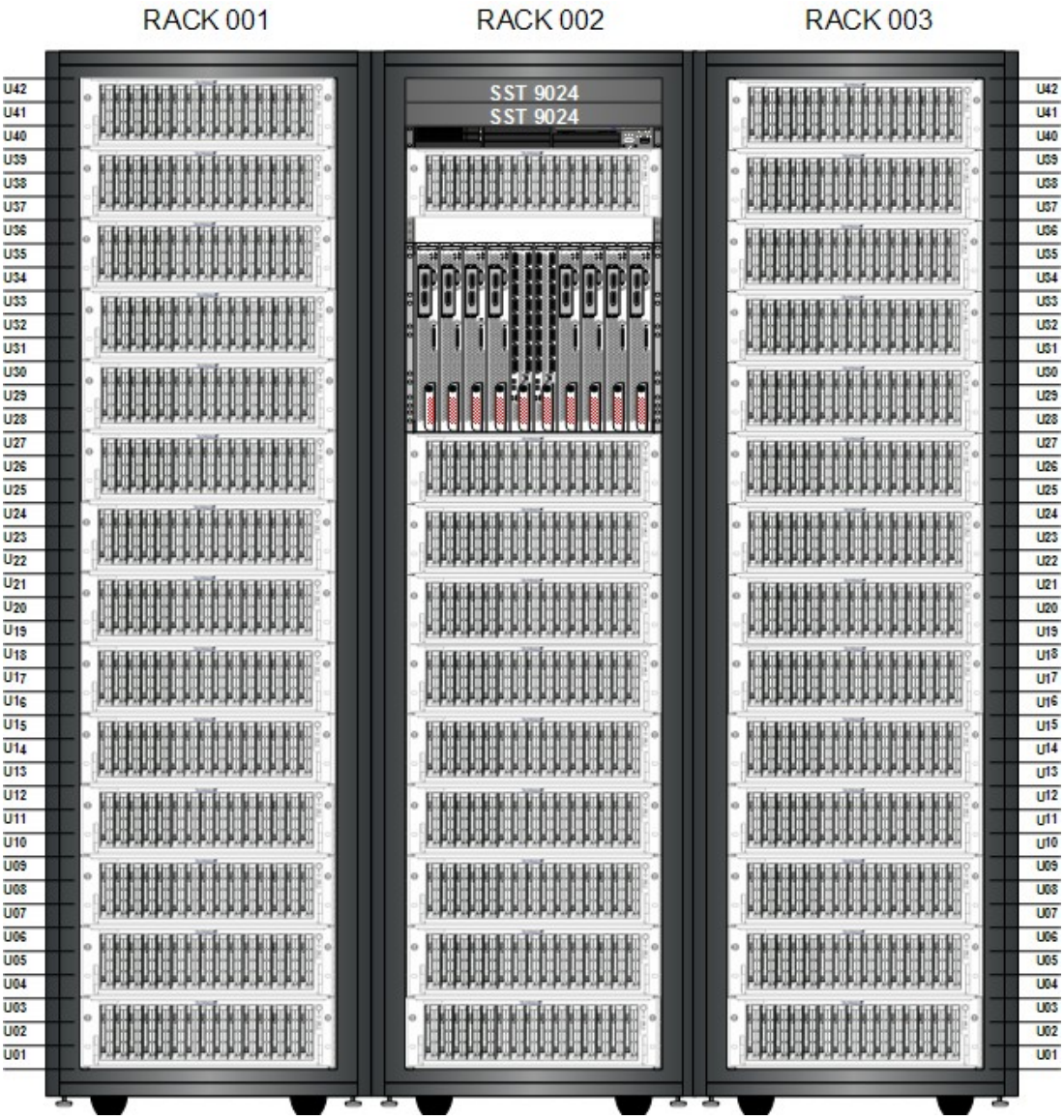


Figure 1.1: PANTAmatrix 64P

2.0 Clause 1: Logical Database Design Related Items

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 build scripts that define, create, and analyze the tables and indices for the TPC-H database.

2.2 Physical Organization

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.

The column reordering of tables used in this benchmark result is described in Appendix B.

2.3 Horizontal Partitioning

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

The horizontal partitioning of tables used in this benchmark result is 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

3.1 Query Language

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

SQL was the query used to implement all queries in this benchmark.

3.2 Verifying Method for 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 2.5.0 of DBGEN and 2.5.0 of QGEN were used in this benchmark

3.3 Generating Values for Substitution Parameters

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.

QGEN version 2.5.0 was used to generate the substitution parameters

3.4 Query Text and Output Data from Qualification 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 definition 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 actual query text and query output.

3.5 Query Substitution Parameters and Seeds Used

The query substitution parameters used for all performance tests must be disclosed in tabular format, along with the seeds used to generate these parameters.

Appendix E contains the seed and query substitution parameters.

3.6 Query Isolation Level

The isolation level used to run the queries must be disclosed. If the isolation level does not map closely

to the 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 Source Code of Refresh Functions

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

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

4.0 Clause 3: Database System Properties

4.1 ACID Properties

The ACID (Atomicity, Consistency, Isolation, and Durability) properties of transaction processing systems must be supported by the system under test during the timed portion of this benchmark. Since TPC-H is not a transaction processing benchmark, the ACID properties must be evaluated outside the timed portion of the test.

The source code for the ACID test is included in Appendix C.

4.2 Atomicity Requirements

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

4.2.1 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 ORDERS, 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.

4.2.2 Atomicity of Aborted Transactions

Perform the ACID Transaction for a randomly selected set of input data, substituting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDERS, 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.

4.3 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 TPCH.LINEITEM WHERE L_ORDERKEY = okey

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

4.3.1 Consistency Test

Verify that ORDERS and LINEITEM tables are initially consistent, submit the prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDERS and LINEITEM.

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 9 execution streams.
3. The consistency of the ORDERS and LINEITEM tables was re-verified.

4.4 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.

4.4.1 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.

4.4.2 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.

4.4.3 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 DELTA1. 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 DELTA2.
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))

4.4.4 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 DELTA1. 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 DELTA2.
3. T2 waited.

4. T1 was allowed to ROLLBACK and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE.

4.4.5 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 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.

4.4.6 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.

1. A Transaction, T1, was started which executed Q1 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 Q1.

4.5 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.

4.5.1 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.

The disks containing TPC-H log files were on RAID 1/0 protected disk groups (see Section 5.8). During the durability test, one disk was removed from the RAID group containing the log. The test continued uninterrupted, because of the RAID protection.

Next, one disk containing TPC-H data files (not duplicated or RAID protected) was removed. As

expected, the database crashed. Upon replacement of the drive, the database rebuilt the data files to a consistent state from the log files. The durability success file and the HISTORY table were compared and the counts matched.

4.5.2 System Crash

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

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

4.5.3 Memory Failure

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

The system crash and memory failure tests were combined as explained in section 4.5.2.

5.0 Clause 4: Scaling and Database Population

5.1 Ending 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	Cardinality
ORDER	1,500,000,000
LINEITEM	5,999,989,709
CUSTOMER	150,000,000
PART	200,000,000
SUPPLIER	10,000,000
PARTSUPP	800,000,000
NATION	25
REGION	5

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 distribution of tables and logs across media are as follows:

The 37 PANTA Systems Storage Array Modules export two types of disks as described in Section 1.3. The 256 2-drive RAID0 disks are partitioned into 3 partitions of sizes 8GB, 20GB, and 20GB. The remaining individual 6 disks are partitioned into 2 partitions of sizes 1GB, 1GB

The first partition of each disk is used for all tables and indexes. The second partition is used for for temp space. The third partition (certain drives only) is used for log, sys, undo and flat files data.

Data Type	Arrays	Disk Position(s)	Disk Type	# Partitions	Partition	Size
Tables + Indexes	1-37	1-6 or 1-7	RAID0	256	1	8GB
Temp Space	1-37	1-6 or 1-7	RAID0	256	2	20GB
Log	1-32	3	RAID0*	32	3	20GB
Undo	odd 1-31	5	RAID0	16	3	20GB
Flat Files	1-32	1, 2, 6, 7	RAID0**	128	3	20GB
Lock	34, 36, 37	7	pass-thru	3	1	1GB
Config	34, 36	8	pass-thru	2	1	1GB
Ctrl	34, 36	8	pass-thru	2	2	1GB
Sys	37	8	pass-thru	1	1	1GB
Aux	37	8	pass-thru	1	2	1GB

Table 5.2: Disk Partition Usage

* The physical log partitions above are mirrored using Linux software RAID1. Physical log partition 1 mirrors physical log partition 17, physical log partition 2 mirrors physical log partition 18, and so on. The result is 16 logical log partitions created from 32 physical partitions. Each physical partition is on a HW RAID0 group spanning 2 actual disks. Thus a total of 64 actual disks are utilized for logs.

** The physical flat file partitions above are additionally striped using Linux software RAID0. Physical flat file partitions 1, 2, 17, 18, 33, 34, 49, 50, 65, 66, 81, 82, 97, 98, 113, 114 are striped to form logical flat file partition 1. Physical flat file partitions 3, 4, 19, 20, 35, 36, 51, 52, 67, 68, 83, 84, 99, 100, 115, 116 are striped to form logical flat file partition 2, and so on. The result is 8 logical flat file partitions created from 128 physical partitions. Each physical partition is on a HW RAID0 group spanning 2 actual disks. Thus a total of 256 actual disks are utilized for flat files.

The OS root and the Oracle home directory were configured on the disks contained in a separate dedicated PANTAmatrix System Storage Module. This dedicated Storage Module is a basic requirement of PANTAmatrix and is used in lieu of the local direct-attached hard drives normally found in traditional servers.

5.3 Mapping of Database Partitions/Replication

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

Horizontal partitioning was used for all base and index tables except NATION and REGION. The details of this partitioning can be understood by examining the syntax of the table and index definition statements in Appendix B. Similar partitioning was used in the qualification database size.

Section 5.2 describes the distribution of tables and logs across all media.

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 must be disclosed for each device.

Hardware RAID0 was used for tables, index, temp, log, undo, and flat files. Additionally, as described in section 5.2, SW RAID1 was used on the log partitions. Also as described in section 5.2, SW RAID0 was further used on top of the HW RAID0 flat file partitions.

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 2.5.0 was not modified to generate the database population for this benchmark.

5.6 Database Load Time

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

A backup of the database partitions is required to meet the durability requirements. The backup time (15 min. 47 sec.) was added into the elapsed load time.

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.

The data storage ratio is computed in Table 5.7.

Disk Type	Number of Disks	Space per Disk	Sub-Total Disk Space	Database Scale Factor	Storage Ratio
PANTA Storage Array Modules (x37)	518	250 GB	129,500.0 GB		
PANTA System Storage Module (x1)	14	250 GB	3,500.0 GB		
Scale Factor			133,000.0 GB	1,000 GB	133.0

Table 5.7: Ratio of Storage Used in the PANTAmatrix Database Storage Subsystem

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.

The database was loaded using data generation stored on the flat files all on the tested and priced configuration. DBGEN was used to create the flat files.

The flat files are stored in 8 filesystems. Each filesystem lies on top of a SW RAID0 device that spans across the third partition on each of 128 HW RAID0 disks spread over 32 of the PANTA System Storage Array Modules. SW RAID striping was done on the nodes using the Linux MD driver. Figure 5.8.1 illustrates the physical location of each disk used for flat file storage.

Data Type	Arrays	Disk Position(s)	Disk Type	# Partitions	Partition	Size
Flat Files	1-32	1, 2, 6, 7	RAID0	64	3	20GB

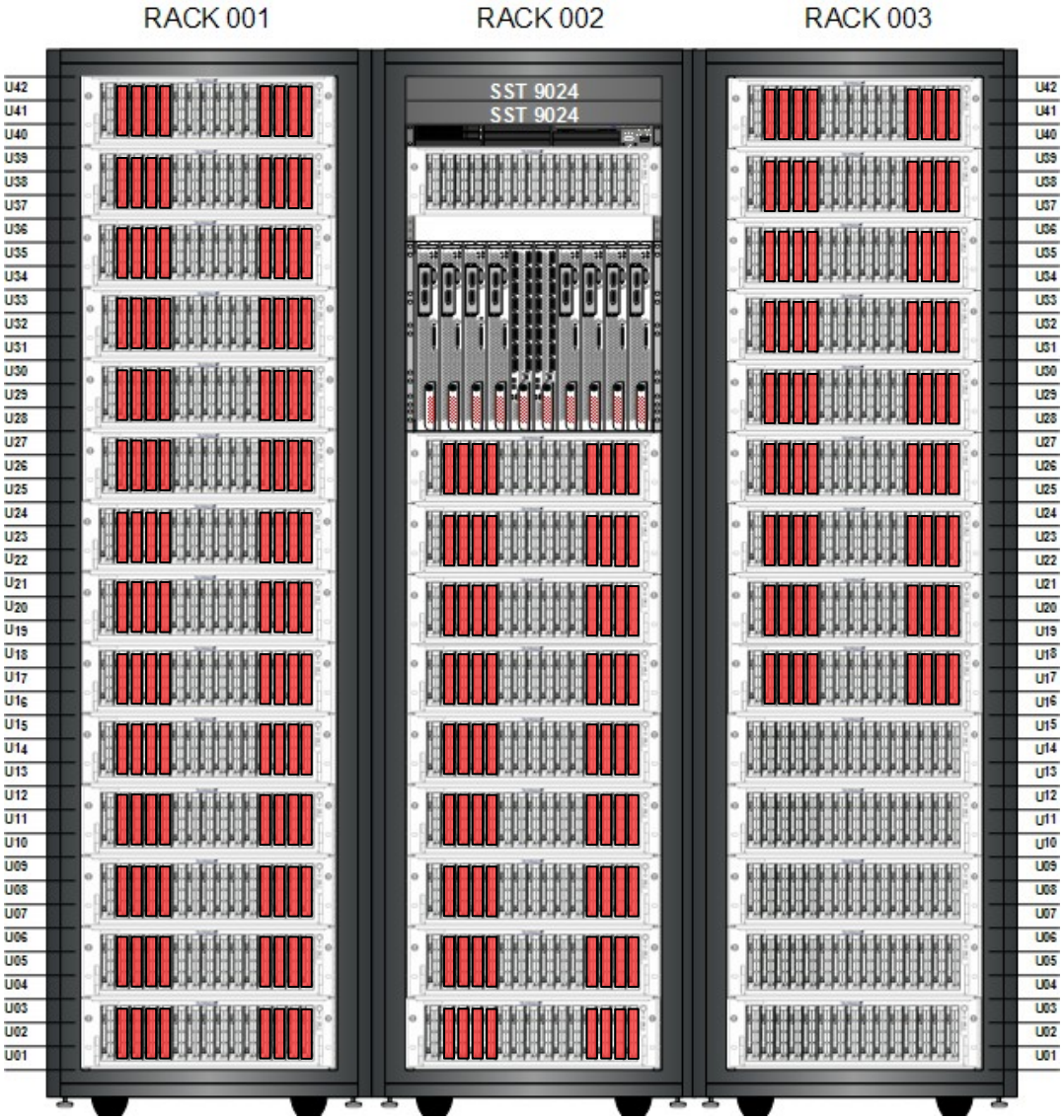


Figure 5.8.1: Disk Partitions Used for Flat File Storage

The block diagram in figure 5.8.2 describes the process used to load the database.

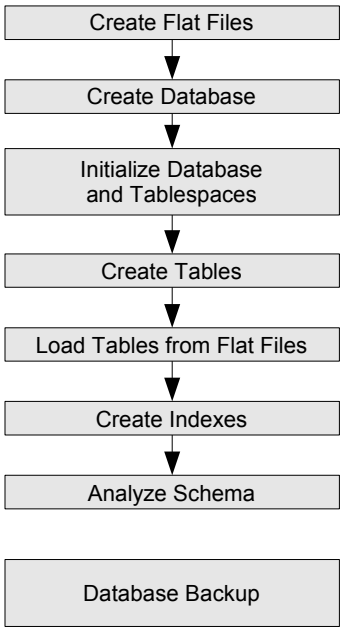


Figure 5.8.2: Database load procedure

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 Referential Integrity of Initial Database Load

A test was performed to show referential integrity of the database after initial load.

5.11 Base Data Comparison

The rows in the database after the performance test were verified against the Base, Insert and Delete reference data sets by extracting 5 random rows from each table small number and comparing them against the corresponding Base, Insert and Delete reference data set files.

6.0 Clause 5: Performance Metrics and Execution Rules Related Items

6.1 System Activity Between Load and Performance Tests

Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed.

A script was run to display the hardware configurations of the SUT. Auditor requested queries were run against the database to verify the correctness of the database load. The database was restarted. All scripts and queries used are included in Appendix E.

6.2 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. Database started
2. RF1 Refresh Transaction
3. Stream 00 Execution
4. RF2 Refresh Transaction

6.3 Timing Intervals for Each Query and Refresh Functions

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

The timing intervals for each query and both update functions are given in the Numerical Quantities Summary earlier in this document.

6.4 Number of Streams for the Throughput Test

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

8 streams were used for the throughput test.

6.5 Start and End Date/Time of Each Query Stream

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

The throughput test start time and finish time for each stream are given in the Numerical Quantities Summary earlier in this document.

6.6 Total Elapsed Time of the Measurement Interval

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

The total elapsed time of the throughput test is given in the Numerical Quantities Summary earlier in this document.

6.7 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.

		RF1		RF2	
Number	Date	Start	End	Start	End
1	10/19/2006	11:00:25	11:01:57	11:01:57	11:04:33
2	10/19/2006	11:04:33	11:05:59	11:05:59	11:08:30
3	10/19/2006	11:08:30	11:09:53	11:09:53	11:12:21
4	10/19/2006	11:12:21	11:13:41	11:13:41	11:16:09
5	10/19/2006	11:16:09	11:17:22	11:17:22	11:19:43
6	10/19/2006	11:19:43	11:20:58	11:20:58	11:23:24
7	10/19/2006	11:23:24	11:24:33	11:24:33	11:26:43
8	10/19/2006	11:26:43	11:27:48	11:27:48	11:29:57

Table 6.7: Start and Finish Times for RF1 and RF2

6.8 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 refresh function must be reported for the throughput test.

The timing intervals for each query and each update function are given in the Numerical Quantities Summary earlier in this document.

6.9 Performance Metrics

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

The performance metrics, and the numbers, on which they are based, is given in the Numerical Quantities Summary earlier in this document.

6.10 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 reproducibility runs.

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

	QppH@1000GB	QthH@1000GB	QphH@1000GB
Reported Run	72,848.5	48,359.0	59,353.9
Reproducibility Run	78,335.2	49,913.3	62,529.7
% Difference	7.0%	3.1%	5.1%

Table 6.10: Performance Results from Reported and Reproducibility Runs

6.11 System Activity Between Performance Tests

Any activity on the SUT that takes place between the conclusion of Run 1 and the beginning of Run 2 must be disclosed.

There was no system activity 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.

All stream executions are performed by a single script. 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 use must be disclosed.

Profile-directed optimization subject to the requirements of 5.2.9 and 5.2.10 was used.

8.0 Clause 7: Pricing

8.1 Hardware and Software Used in the Priced System

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 Three Year Price

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

A detailed pricing sheet of all the hardware and software used in this configuration and the 3-year maintenance costs, demonstrating the computation of the total 3-year price of the configuration, is included in the executive summary at the beginning of this document.

8.3 Availability Date

The committed delivery date for general availability of products used in the priced 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.

The availability date is April 15, 2007

8.4 Country-Specific Pricing

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

The configuration is priced for the United States of America.

9.0 Clause 8: Auditor's Information and Attestation Letter

9.1 Auditor's 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.

This implementation of the TPC Benchmark H was audited by Francois Raab for InfoSizing.

Further information regarding the audit process may be obtained from:

Francois Raab
InfoSizing, Inc.
1373 N. Franklin Steet
Colorado Springs, CO 80903
(719) 473-7555
(719) 473-7554

Email: francois@infosizing.com

The auditor's attestation letter is included below.

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

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: Richard Au
PANTA Systems
2901 Patrick Henry Drive
Santa Clara, CA 95054

October 22, 2006

I verified the TPC Benchmark™ H performance of the following configuration:

Platform: **PANTA Systems PANTAmatrix**

Database Manager: **Oracle Database 10g Enterprise Edition R2 with RAC and Partitioning**

Operating System: **Red Hat Enterprise Linux 4 Advanced Server Update 3**

The results were:

CPU (Speed)	Memory	Disks	**QphH@1000GB**
PANTA Systems PANTAmatrix (8 nodes, each with)			
4 x AMD Opteron 875HE Dual-Core (2.2GHz)	1 MB Cache/core 32 GB Main	518 x 250 GB 7200rpm SATA.	**59,353.9**

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

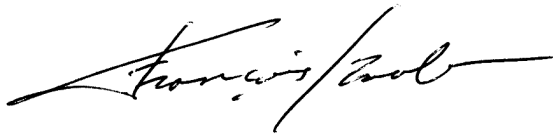
- The database records were defined with the proper layout and size
- The database population was generated using DBGEN and verified by sampling
- The database was properly scaled to1,000GB and populated accordingly
- The compliance of the database auxiliary data structures and referential integrity was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met

- The query input variables were generated by QGEN
- The query text was produced using minor modifications and one query variant
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 8 query streams (a minimum of 7 streams is required)
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified
- The required amount of database log was configured
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,



François Raab
President

10.0 Report Availability

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

Transaction Processing Performance Council
Presidio of San Francisco
Building 572B (surface)
P.O. Box 29920 (mail)
San Francisco, CA 94129-0920
USA

Telephone: (415) 561-6272
Fax: (415) 561-6120
Email: info@tpc.org

Appendix A: System and Database Configurations and Parameters

```
# ---------------------------------------------------------------
# init_run.ora
# ---------------------------------------------------------------

aq_tm_processes                 = 0
audit_trail                     = false
compatible                      = 10.2.0.2.0
control_files                   =
(/home/oracle/partitions/ctrl_1,/home/oracle/partitions/ctrl_2)
db_block_checksum               = false
db_block_size                   = 16384
db_cache_size                   = 15g
db_file_multiblock_read_count   = 64
db_files                        = 500
db_name                         = 10i
db_writer_processes             = 4
dml_locks                       = 400000
global_names                    = false
instance_name                   = raca
log_buffer                      = 16777216
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            = 9g
processes                       = 1000
recovery_parallelism            = 8
replication_dependency_tracking = false
statistics_level                = basic
undo_retention                  = 400000
optimizer_index_cost_adj        = 1400
cpu_count                       = 8
shared_pool_size                = 4g
UNDO_MANAGEMENT                 = AUTO
parallel_threads_per_cpu        = 2


# ---------------------------------------------------------------
# init_tpch1.ora
# ---------------------------------------------------------------

instance_number = 1
thread = 1
undo_management = auto
undo_tablespace = ts_undo1
cluster_database = true
cluster_interconnects = 10.6.128.21
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch2.ora
# ---------------------------------------------------------------

instance_number = 2
thread = 2
undo_management = auto
undo_tablespace = ts_undo2
cluster_database = true
cluster_interconnects = 10.6.128.22
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch3.ora
# ---------------------------------------------------------------

instance_number = 3
thread = 3
undo_management = auto
undo_tablespace = ts_undo3
cluster_database = true
cluster_interconnects = 10.6.128.23
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch4.ora
# ---------------------------------------------------------------

instance_number = 4
thread = 4
undo_management = auto
undo_tablespace = ts_undo4
cluster_database = true
cluster_interconnects = 10.6.128.24
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch5.ora
# ---------------------------------------------------------------

instance_number = 5
thread = 5
undo_management = auto
undo_tablespace = ts_undo5
cluster_database = true
cluster_interconnects = 10.6.128.25
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch6.ora
# ---------------------------------------------------------------

instance_number = 6
thread = 6
undo_management = auto
undo_tablespace = ts_undo6
cluster_database = true
cluster_interconnects = 10.6.128.26
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch7.ora
# ---------------------------------------------------------------

instance_number = 7
thread = 7
undo_management = auto
undo_tablespace = ts_undo7
cluster_database = true
cluster_interconnects = 10.6.128.27
ifile = /home/oracle/kit/init_run.ora


# ---------------------------------------------------------------
# init_tpch8.ora
# ---------------------------------------------------------------

instance_number = 8
thread = 8
undo_management = auto
undo_tablespace = ts_undo8
cluster_database = true
cluster_interconnects = 10.6.128.28
ifile = /home/oracle/kit/init_run.ora
```

Appendix B: Database Build Scripts

```
# --------------------------------------------------------------------
# adddf.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_nasm.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/partitions/undo_${1}' size 2048m reuse;
alter database add logfile thread ${1}
'/home/oracle/partitions/log_${one}' size 4096m reuse,
'/home/oracle/partitions/log_${two}' size 4096m reuse;
alter database enable public thread ${1};
!
echo end creating undo and log for node $1 `date`


# --------------------------------------------------------------------
# cr_ts_p.sh
# --------------------------------------------------------------------

sqlplus /NOLOG <<!!
connect /as sysdba
drop tablespace $1 including contents;
create tablespace $1 nologging
datafile '$2' size $3 reuse extent management dictionary default
storage (initial 10m next 10m maxextents unlimited pctincrease 0);
!!


# --------------------------------------------------------------------
# dapop_final.sh
# --------------------------------------------------------------------

#!/bin/bash
sqlplus /NOLOG <<EOF
connect / as sysdba
drop user tpch cascade;
grant DBA
to tpch identified by tpch;
connect tpch/tpch;
drop directory ff1;
create directory ff1 as '/data01';
drop directory ff2;
create directory ff2 as '/data02';
drop directory ff3;
create directory ff3 as '/data03';
drop directory ff4;
create directory ff4 as '/data04';
drop directory ff5;
create directory ff5 as '/data05';
drop directory ff6;
create directory ff6 as '/data06';
drop directory ff7;
create directory ff7 as '/data07';
drop directory ff8;
create directory ff8 as '/data08';

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',
ff1:'lineitem.tbl.2',
ff1:'lineitem.tbl.3',
ff1:'lineitem.tbl.4',
ff1:'lineitem.tbl.5',
ff1:'lineitem.tbl.6',
ff1:'lineitem.tbl.7',
ff1:'lineitem.tbl.8',
ff2:'lineitem.tbl.9',
ff2:'lineitem.tbl.10',
ff2:'lineitem.tbl.11',
ff2:'lineitem.tbl.12',
ff2:'lineitem.tbl.13',
ff2:'lineitem.tbl.14',
ff2:'lineitem.tbl.15',
ff2:'lineitem.tbl.16',
ff3:'lineitem.tbl.17',
ff3:'lineitem.tbl.18',
ff3:'lineitem.tbl.19',
ff3:'lineitem.tbl.20',
ff3:'lineitem.tbl.21',
ff3:'lineitem.tbl.22',
ff3:'lineitem.tbl.23',
ff3:'lineitem.tbl.24',
ff4:'lineitem.tbl.25',
ff4:'lineitem.tbl.26',
ff4:'lineitem.tbl.27',
ff4:'lineitem.tbl.28',
ff4:'lineitem.tbl.29',
ff4:'lineitem.tbl.30',
ff4:'lineitem.tbl.31',
ff4:'lineitem.tbl.32',
ff5:'lineitem.tbl.33',
ff5:'lineitem.tbl.34',
ff5:'lineitem.tbl.35',
ff5:'lineitem.tbl.36',
ff5:'lineitem.tbl.37',
ff5:'lineitem.tbl.38',
ff5:'lineitem.tbl.39',
ff5:'lineitem.tbl.40',
ff6:'lineitem.tbl.41',
ff6:'lineitem.tbl.42',
ff6:'lineitem.tbl.43',
ff6:'lineitem.tbl.44',
ff6:'lineitem.tbl.45',
ff6:'lineitem.tbl.46',
ff6:'lineitem.tbl.47',
ff6:'lineitem.tbl.48',
ff7:'lineitem.tbl.49',
ff7:'lineitem.tbl.50',
ff7:'lineitem.tbl.51',
ff7:'lineitem.tbl.52',
ff7:'lineitem.tbl.53',
ff7:'lineitem.tbl.54',
ff7:'lineitem.tbl.55',
ff7:'lineitem.tbl.56',
ff8:'lineitem.tbl.57',
ff8:'lineitem.tbl.58',
ff8:'lineitem.tbl.59',
ff8:'lineitem.tbl.60',
ff8:'lineitem.tbl.61',
ff8:'lineitem.tbl.62',
ff8:'lineitem.tbl.63'
))
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',
ff1:'orders.tbl.2',
ff1:'orders.tbl.3',
ff1:'orders.tbl.4',
ff1:'orders.tbl.5',
ff1:'orders.tbl.6',
ff1:'orders.tbl.7',
ff1:'orders.tbl.8',
ff2:'orders.tbl.9',
ff2:'orders.tbl.10',
ff2:'orders.tbl.11',
ff2:'orders.tbl.12',
ff2:'orders.tbl.13',
ff2:'orders.tbl.14',
ff2:'orders.tbl.15',
ff2:'orders.tbl.16',
ff3:'orders.tbl.17',
ff3:'orders.tbl.18',
ff3:'orders.tbl.19',
ff3:'orders.tbl.20',
ff3:'orders.tbl.21',
ff3:'orders.tbl.22',
ff3:'orders.tbl.23',
ff3:'orders.tbl.24',
ff4:'orders.tbl.25',
ff4:'orders.tbl.26',
ff4:'orders.tbl.27',
ff4:'orders.tbl.28',
ff4:'orders.tbl.29',
ff4:'orders.tbl.30',
ff4:'orders.tbl.31',
ff4:'orders.tbl.32',
ff5:'orders.tbl.33',
ff5:'orders.tbl.34',
ff5:'orders.tbl.35',
ff5:'orders.tbl.36',
ff5:'orders.tbl.37',
ff5:'orders.tbl.38',
ff5:'orders.tbl.39',
ff5:'orders.tbl.40',
ff6:'orders.tbl.41',
ff6:'orders.tbl.42',
ff6:'orders.tbl.43',
ff6:'orders.tbl.44',
ff6:'orders.tbl.45',
ff6:'orders.tbl.46',
ff6:'orders.tbl.47',
ff6:'orders.tbl.48',
ff7:'orders.tbl.49',
ff7:'orders.tbl.50',
ff7:'orders.tbl.51',
ff7:'orders.tbl.52',
ff7:'orders.tbl.53',
ff7:'orders.tbl.54',
ff7:'orders.tbl.55',
ff7:'orders.tbl.56',
ff8:'orders.tbl.57',
ff8:'orders.tbl.58',
ff8:'orders.tbl.59',
ff8:'orders.tbl.60',
ff8:'orders.tbl.61',
ff8:'orders.tbl.62',
ff8:'orders.tbl.63'
))
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',
ff1:'partsupp.tbl.2',
ff1:'partsupp.tbl.3',
ff1:'partsupp.tbl.4',
ff1:'partsupp.tbl.5',
ff1:'partsupp.tbl.6',
ff1:'partsupp.tbl.7',
ff1:'partsupp.tbl.8',
ff2:'partsupp.tbl.9',
ff2:'partsupp.tbl.10',
ff2:'partsupp.tbl.11',
ff2:'partsupp.tbl.12',
ff2:'partsupp.tbl.13',
ff2:'partsupp.tbl.14',
ff2:'partsupp.tbl.15',
ff2:'partsupp.tbl.16',
ff3:'partsupp.tbl.17',
ff3:'partsupp.tbl.18',
ff3:'partsupp.tbl.19',
ff3:'partsupp.tbl.20',
ff3:'partsupp.tbl.21',
ff3:'partsupp.tbl.22',
ff3:'partsupp.tbl.23',
ff3:'partsupp.tbl.24',
ff4:'partsupp.tbl.25',
ff4:'partsupp.tbl.26',
ff4:'partsupp.tbl.27',
ff4:'partsupp.tbl.28',
ff4:'partsupp.tbl.29',
ff4:'partsupp.tbl.30',
ff4:'partsupp.tbl.31',
ff4:'partsupp.tbl.32',
ff5:'partsupp.tbl.33',
ff5:'partsupp.tbl.34',
ff5:'partsupp.tbl.35',
ff5:'partsupp.tbl.36',
ff5:'partsupp.tbl.37',
ff5:'partsupp.tbl.38',
ff5:'partsupp.tbl.39',
ff5:'partsupp.tbl.40',
ff6:'partsupp.tbl.41',
ff6:'partsupp.tbl.42',
ff6:'partsupp.tbl.43',
ff6:'partsupp.tbl.44',
ff6:'partsupp.tbl.45',
ff6:'partsupp.tbl.46',
ff6:'partsupp.tbl.47',
ff6:'partsupp.tbl.48',
ff7:'partsupp.tbl.49',
ff7:'partsupp.tbl.50',
ff7:'partsupp.tbl.51',
ff7:'partsupp.tbl.52',
ff7:'partsupp.tbl.53',
ff7:'partsupp.tbl.54',
ff7:'partsupp.tbl.55',
ff7:'partsupp.tbl.56',
ff8:'partsupp.tbl.57',
ff8:'partsupp.tbl.58',
ff8:'partsupp.tbl.59',
ff8:'partsupp.tbl.60',
ff8:'partsupp.tbl.61',
ff8:'partsupp.tbl.62',
ff8:'partsupp.tbl.63'
))
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',
ff1:'part.tbl.2',
ff1:'part.tbl.3',
ff1:'part.tbl.4',
ff1:'part.tbl.5',
ff1:'part.tbl.6',
ff1:'part.tbl.7',
ff1:'part.tbl.8',
```

```
ff2:'part.tbl.9',
ff2:'part.tbl.10',
ff2:'part.tbl.11',
ff2:'part.tbl.12',
ff2:'part.tbl.13',
ff2:'part.tbl.14',
ff2:'part.tbl.15',
ff2:'part.tbl.16',
ff3:'part.tbl.17',
ff3:'part.tbl.18',
ff3:'part.tbl.19',
ff3:'part.tbl.20',
ff3:'part.tbl.21',
ff3:'part.tbl.22',
ff3:'part.tbl.23',
ff3:'part.tbl.24',
ff4:'part.tbl.25',
ff4:'part.tbl.26',
ff4:'part.tbl.27',
ff4:'part.tbl.28',
ff4:'part.tbl.29',
ff4:'part.tbl.30',
ff4:'part.tbl.31',
ff4:'part.tbl.32',
ff5:'part.tbl.33',
ff5:'part.tbl.34',
ff5:'part.tbl.35',
ff5:'part.tbl.36',
ff5:'part.tbl.37',
ff5:'part.tbl.38',
ff5:'part.tbl.39',
ff5:'part.tbl.40',
ff6:'part.tbl.41',
ff6:'part.tbl.42',
ff6:'part.tbl.43',
ff6:'part.tbl.44',
ff6:'part.tbl.45',
ff6:'part.tbl.46',
ff6:'part.tbl.47',
ff6:'part.tbl.48',
ff7:'part.tbl.49',
ff7:'part.tbl.50',
ff7:'part.tbl.51',
ff7:'part.tbl.52',
ff7:'part.tbl.53',
ff7:'part.tbl.54',
ff7:'part.tbl.55',
ff7:'part.tbl.56',
ff8:'part.tbl.57',
ff8:'part.tbl.58',
ff8:'part.tbl.59',
ff8:'part.tbl.60',
ff8:'part.tbl.61',
ff8:'part.tbl.62',
ff8:'part.tbl.63'
))
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',
ff1:'customer.tbl.2',
ff1:'customer.tbl.3',
ff1:'customer.tbl.4',
ff1:'customer.tbl.5',
ff1:'customer.tbl.6',
ff1:'customer.tbl.7',
ff1:'customer.tbl.8',
ff2:'customer.tbl.9',
ff2:'customer.tbl.10',
ff2:'customer.tbl.11',
ff2:'customer.tbl.12',
ff2:'customer.tbl.13',
ff2:'customer.tbl.14',
ff2:'customer.tbl.15',
ff2:'customer.tbl.16',
ff3:'customer.tbl.17',
ff3:'customer.tbl.18',
ff3:'customer.tbl.19',
ff3:'customer.tbl.20',
ff3:'customer.tbl.21',
ff3:'customer.tbl.22',
ff3:'customer.tbl.23',
ff3:'customer.tbl.24',
ff4:'customer.tbl.25',
ff4:'customer.tbl.26',
ff4:'customer.tbl.27',
ff4:'customer.tbl.28',
ff4:'customer.tbl.29',
ff4:'customer.tbl.30',
ff4:'customer.tbl.31',
ff4:'customer.tbl.32',
ff5:'customer.tbl.33',
ff5:'customer.tbl.34',
ff5:'customer.tbl.35',
ff5:'customer.tbl.36',
ff5:'customer.tbl.37',
ff5:'customer.tbl.38',
ff5:'customer.tbl.39',
ff5:'customer.tbl.40',
ff6:'customer.tbl.41',
ff6:'customer.tbl.42',
ff6:'customer.tbl.43',
ff6:'customer.tbl.44',
ff6:'customer.tbl.45',
ff6:'customer.tbl.46',
ff6:'customer.tbl.47',
ff6:'customer.tbl.48',
ff7:'customer.tbl.49',
ff7:'customer.tbl.50',
ff7:'customer.tbl.51',
ff7:'customer.tbl.52',
ff7:'customer.tbl.53',
ff7:'customer.tbl.54',
ff7:'customer.tbl.55',
ff7:'customer.tbl.56',
ff8:'customer.tbl.57',
ff8:'customer.tbl.58',
ff8:'customer.tbl.59',
ff8:'customer.tbl.60',
ff8:'customer.tbl.61',
ff8:'customer.tbl.62',
ff8:'customer.tbl.63'
))
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',
ff1:'supplier.tbl.2',
ff1:'supplier.tbl.3',
ff1:'supplier.tbl.4',
ff1:'supplier.tbl.5',
ff1:'supplier.tbl.6',
ff1:'supplier.tbl.7',
ff1:'supplier.tbl.8',
ff2:'supplier.tbl.9',
ff2:'supplier.tbl.10',
ff2:'supplier.tbl.11',
ff2:'supplier.tbl.12',
ff2:'supplier.tbl.13',
ff2:'supplier.tbl.14',
ff2:'supplier.tbl.15',
ff2:'supplier.tbl.16',
ff3:'supplier.tbl.17',
ff3:'supplier.tbl.18',
ff3:'supplier.tbl.19',
ff3:'supplier.tbl.20',
ff3:'supplier.tbl.21',
ff3:'supplier.tbl.22',
ff3:'supplier.tbl.23',
ff3:'supplier.tbl.24',
ff4:'supplier.tbl.25',
ff4:'supplier.tbl.26',
ff4:'supplier.tbl.27',
```

```
ff4:'supplier.tbl.28',
ff4:'supplier.tbl.29',
ff4:'supplier.tbl.30',
ff4:'supplier.tbl.31',
ff4:'supplier.tbl.32',
ff5:'supplier.tbl.33',
ff5:'supplier.tbl.34',
ff5:'supplier.tbl.35',
ff5:'supplier.tbl.36',
ff5:'supplier.tbl.37',
ff5:'supplier.tbl.38',
ff5:'supplier.tbl.39',
ff5:'supplier.tbl.40',
ff6:'supplier.tbl.41',
ff6:'supplier.tbl.42',
ff6:'supplier.tbl.43',
ff6:'supplier.tbl.44',
ff6:'supplier.tbl.45',
ff6:'supplier.tbl.46',
ff6:'supplier.tbl.47',
ff6:'supplier.tbl.48',
ff7:'supplier.tbl.49',
ff7:'supplier.tbl.50',
ff7:'supplier.tbl.51',
ff7:'supplier.tbl.52',
ff7:'supplier.tbl.53',
ff7:'supplier.tbl.54',
ff7:'supplier.tbl.55',
ff7:'supplier.tbl.56',
ff8:'supplier.tbl.57',
ff8:'supplier.tbl.58',
ff8:'supplier.tbl.59',
ff8:'supplier.tbl.60',
ff8:'supplier.tbl.61',
ff8:'supplier.tbl.62',
ff8:'supplier.tbl.63'
))
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;

alter table l_et parallel 128;
alter table o_et parallel 128;
alter table ps_et parallel 128;
alter table p_et parallel 128;
alter table c_et parallel 128;
alter table s_et parallel 128;

alter user tpch default tablespace ts_tsd1;
alter user tpch temporary tablespace ts_temp;

@?/rdbms/admin/utlxplan.sql;

set timing on
set echo on
```

```
!date
rem drop table orders;
create table orders(
    o_orderdate          ,
    o_orderkey           NOT NULL,
    o_custkey            NOT NULL,
    o_orderpriority      ,
    o_shippriority       ,
    o_clerk              ,
    o_orderstatus        ,
    o_totalprice         ,
    o_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 10m next 10m freelist groups 8 freelists 99)
compress
parallel 128
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 128
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
```

```
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
```

```
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
```

```
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
```

```
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
```

```
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
```

```
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition ord84 values less than (MAXVALUE)
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
)
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

--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 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
storage (initial 10m next 10m freelist groups 8 freelists 99)
parallel 128
nologging
as select
    ps_partkey            ,
    ps_suppkey            ,
    ps_supplycost         ,
    ps_availqty           ,
    ps_comment
from ps_et;
!date

--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 10m next 10m freelist groups 8 freelists 99)
compress
parallel 128
nologging
partition by hash (c_custkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
```

```
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
as select
    c_custkey           ,
    c_mktsegment        ,
    c_nationkey         ,
    c_name              ,
    c_address           ,
    c_phone             ,
    c_acctbal           ,
    c_comment
from c_et;
!date

--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 10m next 10m freelist groups 8 freelists 99)
compress
parallel 128
nologging
partition by hash (p_partkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
as select
    p_partkey           ,
    p_type              ,
    p_size              ,
    p_brand             ,
    p_name              ,
    p_container         ,
    p_mfgr              ,
    p_retailprice       ,
    p_comment
from p_et;
!date

--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 10m next 10m freelist groups 8 freelists 99)
compress
parallel 128
nologging
partition by hash (s_suppkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
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

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 10m next 10m freelist groups 8 freelists 99)
compress
parallel 128
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 128
(
partition item1 values less than (to_date('1992-01-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item2 values less than (to_date('1992-02-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item3 values less than (to_date('1992-03-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item4 values less than (to_date('1992-04-01','YYYY-MM-DD'))
```

```
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item5 values less than (to_date('1992-05-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item6 values less than (to_date('1992-06-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item7 values less than (to_date('1992-07-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item8 values less than (to_date('1992-08-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item9 values less than (to_date('1992-09-01','YYYY-MM-DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item17 values less than (to_date('1993-05-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
```

```
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item24 values less than (to_date('1993-12-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item25 values less than (to_date('1994-01-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item26 values less than (to_date('1994-02-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item27 values less than (to_date('1994-03-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item28 values less than (to_date('1994-04-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item29 values less than (to_date('1994-05-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item30 values less than (to_date('1994-06-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
```

```
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item31 values less than (to_date('1994-07-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item32 values less than (to_date('1994-08-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item33 values less than (to_date('1994-09-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item34 values less than (to_date('1994-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item35 values less than (to_date('1994-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item36 values less than (to_date('1994-12-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item37 values less than (to_date('1995-01-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item38 values less than (to_date('1995-02-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item39 values less than (to_date('1995-03-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item40 values less than (to_date('1995-04-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item41 values less than (to_date('1995-05-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item42 values less than (to_date('1995-06-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item43 values less than (to_date('1995-07-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
```

```
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item44 values less than (to_date('1995-08-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item45 values less than (to_date('1995-09-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item46 values less than (to_date('1995-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item47 values less than (to_date('1995-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item48 values less than (to_date('1995-12-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item49 values less than (to_date('1996-01-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item50 values less than (to_date('1996-02-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item51 values less than (to_date('1996-03-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
```

```
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
```

```
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
```

```
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128),
partition item84 values less than (MAXVALUE)
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
)
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

drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey) global partition by hash (l_orderkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
pctfree 2
initrans 10
storage (initial 10m next 10m freelist groups 8 freelists 99)
parallel 128
compute statistics
nologging;
alter index i_l_orderkey allocate extent (size 132m);
alter index i_l_orderkey allocate extent (size 132m);
alter index i_l_orderkey allocate extent (size 132m);
alter index i_l_orderkey allocate extent (size 132m);

drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey) global partition by hash (o_orderkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
pctfree 2
initrans 10
storage (initial 10m next 10m freelist groups 8 freelists 99 )
parallel 128
compute statistics
nologging;
alter index i_o_orderkey allocate extent (size 16m);
alter index i_o_orderkey allocate extent (size 16m);
alter index i_o_orderkey allocate extent (size 16m);
alter index i_o_orderkey allocate extent (size 16m);

!date

drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)  global partition by hash (c_custkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
pctfree 2
initrans 10
storage (initial 10m next 10m freelist groups 8 freelists 99)
parallel 128
compute statistics
nologging;
!date

drop index i_ps_partkey_suppkey;
create unique index i_ps_partkey_suppkey
on partsupp (ps_partkey,ps_suppkey)  global partition by hash
(ps_partkey)
partitions 128
store in ( tsd1,tsd2,tsd3,tsd4,tsd5,tsd6,tsd7,tsd8,tsd9,tsd10,tsd11,
tsd12,tsd13,tsd14,tsd15,tsd16,tsd17,tsd18,tsd19,tsd20,tsd21,tsd22,
tsd23,tsd24,tsd25,tsd26,tsd27,tsd28,tsd29,tsd30,tsd31,tsd32,tsd33,
tsd34,tsd35,tsd36,tsd37,tsd38,tsd39,tsd40,tsd41,tsd42,tsd43,tsd44,
tsd45,tsd46,tsd47,tsd48,tsd49,tsd50,tsd51,tsd52,tsd53,tsd54,tsd55,
tsd56,tsd57,tsd58,tsd59,tsd60,tsd61,tsd62,tsd63,tsd64,tsd65,tsd66,
tsd67,tsd68,tsd69,tsd70,tsd71,tsd72,tsd73,tsd74,tsd75,tsd76,tsd77,
tsd78,tsd79,tsd80,tsd81,tsd82,tsd83,tsd84,tsd85,tsd86,tsd87,tsd88,
tsd89,tsd90,tsd91,tsd92,tsd93,tsd94,tsd95,tsd96,tsd97,tsd98,tsd99,
tsd100,tsd101,tsd102,tsd103,tsd104,tsd105,tsd106,tsd107,tsd108,
tsd109,tsd110,tsd111,tsd112,tsd113,tsd114,tsd115,tsd116,tsd117,
tsd118,tsd119,tsd120,tsd121,tsd122,tsd123,tsd124,tsd125,tsd126,
tsd127, tsd128)
pctfree 2
initrans 10
storage (initial 10m next 10m freelist groups 8 freelists 99)
parallel 128
compute statistics
nologging;

!date
set timing on
execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent =>
1, degree => 128, 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

connect tpch/tpch

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

drop directory ff1;
create directory ff1 as '/data01';
drop directory ff2;
create directory ff2 as '/data02';
drop directory ff3;
create directory ff3 as '/data03';
drop directory ff4;
create directory ff4 as '/data04';
drop directory ff5;
create directory ff5 as '/data05';
drop directory ff6;
create directory ff6 as '/data06';
```

```
drop directory ff7;
create directory ff7 as '/data07';
drop directory ff8;
create directory ff8 as '/data08';

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 (
ff8:'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 (
ff8:'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 (
ff8:'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 (
ff8:'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 (
ff8:'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 (
ff8:'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;

alter table l_et parallel 128;
alter table o_et parallel 128;
alter table ps_et parallel 128;
alter table p_et parallel 128;
alter table c_et parallel 128;
alter table s_et parallel 128;

alter session force parallel dml parallel (degree 128);


set timing on
set echo on
!date

insert into partsupp (select
    ps_partkey           ,
    ps_suppkey           ,
    ps_supplycost        ,
    ps_availqty          ,
    ps_comment
from ps_et);
!date

insert into customer (select
    c_custkey            ,
    c_mktsegment         ,
    c_nationkey          ,
    c_name               ,
    c_address            ,
    c_phone              ,
    c_acctbal            ,
    c_comment
from c_et);
!date

insert into part (select
    p_partkey            ,
    p_type               ,
    p_size               ,
    p_brand              ,
    p_name               ,
    p_container          ,
    p_mfgr               ,
    p_retailprice        ,
    p_comment
from p_et);
!date

insert into supplier (select
    s_suppkey            ,
    s_nationkey          ,
    s_comment            ,
    s_name               ,
    s_address            ,
    s_phone              ,
    s_acctbal
from s_et);
!date

commit;

insert into orders (select
    o_orderdate          ,
    o_orderkey           ,
    o_custkey            ,
    o_orderpriority      ,
    o_shippriority       ,
    o_clerk              ,
    o_orderstatus        ,
    o_totalprice         ,
    o_comment
from o_et);
!date

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_et);
commit;
!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

EOF


# ------------------------------------------------------------------
# dapop_final2.sh
# ------------------------------------------------------------------

#!/bin/bash
sqlplus /NOLOG <<EOF
connect tpch/tpch

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

drop directory ff1;
create directory ff1 as '/data01';
drop directory ff2;
create directory ff2 as '/data02';
drop directory ff3;
create directory ff3 as '/data03';
drop directory ff4;
create directory ff4 as '/data04';
drop directory ff5;
create directory ff5 as '/data05';
drop directory ff6;
create directory ff6 as '/data06';
drop directory ff7;
create directory ff7 as '/data07';
drop directory ff8;
create directory ff8 as '/data08';

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 (
```

```
ff8:'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 (
ff8:'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 (
ff8:'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 (
ff8:'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 (
ff8:'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 (
ff8:'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;

alter table l_et parallel 128;
alter table o_et parallel 128;
alter table ps_et parallel 128;
alter table p_et parallel 128;
alter table c_et parallel 128;
alter table s_et parallel 128;

alter session force parallel dml parallel (degree 128);


set timing on
set echo on
!date
```

```
insert into partsupp (select
    ps_partkey          ,
    ps_suppkey          ,
    ps_supplycost       ,
    ps_availqty         ,
    ps_comment
from ps_et);
!date

insert into customer (select
    c_custkey           ,
    c_mktsegment        ,
    c_nationkey         ,
    c_name              ,
    c_address           ,
    c_phone             ,
    c_acctbal           ,
    c_comment
from c_et);
!date

insert into part (select
    p_partkey           ,
    p_type              ,
    p_size              ,
    p_brand             ,
    p_name              ,
    p_container         ,
    p_mfgr              ,
    p_retailprice       ,
    p_comment
from p_et);
!date

insert into supplier (select
    s_suppkey           ,
    s_nationkey         ,
    s_comment           ,
    s_name              ,
    s_address           ,
    s_phone             ,
    s_acctbal
from s_et);
!date

commit;

insert into orders (select
    o_orderdate         ,
    o_orderkey          ,
    o_custkey           ,
    o_orderpriority     ,
    o_shippriority      ,
    o_clerk             ,
    o_orderstatus       ,
    o_totalprice        ,
    o_comment
from o_et);
!date

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_et);
commit;
!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

EOF


# ----------------------------------------------------------------
# dbcre_final.sh
# ----------------------------------------------------------------

#!/bin/ksh
echo "database creation"
date;

sqlplus  /NOLOG <<!
connect /as sysdba
shutdown abort
startup pfile = /home/oracle/kit/init_run.ora nomount;
create database
  controlfile reuse
  logfile '/home/oracle/partitions/log_1'  size 4096m reuse,
          '/home/oracle/partitions/log_2'  size 4096m reuse
  datafile '/home/oracle/partitions/sys' size 2048m reuse
  sysaux datafile '/home/oracle/partitions/aux' size 2048m reuse
  undo tablespace ts_undo1 datafile '/home/oracle/partitions/undo_1'
size 2048m reuse
  default temporary tablespace ts_temp
    tempfile '/home/oracle/partitions/temp_1' size 20000m reuse
    extent management local uniform size 50m
  maxlogfiles 128
  maxdatafiles 4000
  maxinstances 16;
!

(( n=1 ))
while (( n<8 ));do
  (( n=n+1 ))
  /home/oracle/kit/audit/addundolog_nasm.sh $n &
done
wait

sqlplus  /NOLOG <<!
connect /as sysdba
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
!


# ----------------------------------------------------------------
# tscre_final.sh
# ----------------------------------------------------------------

#!/bin/ksh
echo "START: tablespace creation"
i=0
while [ $i -lt 128 ]
do
        i=`expr $i + 1`
        l=`expr $i `
        /home/oracle/kit/audit/cr_ts_p.sh tsd${i}
/home/oracle/partitions/data_${i} 8000m &
done
wait
i=0
while [ $i -lt 128 ]
do
        i=`expr $i + 1`
        j=`expr $i + 128`
        /home/oracle/kit/audit/adddf.sh tsd${i}
/home/oracle/partitions/data_${j} 8000m &
done
wait


i=1
while [ $i -lt 256 ]
do
        i=`expr $i + 1`
        /home/oracle/kit/audit/addtts.sh ts_temp
/home/oracle/partitions/temp_${i} 20000m
done
exit;
```

Appendix C: ACID Scripts

```
# ------------------------------------------------------------
# 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
#

. /home/oracle/acid/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`..."


# ------------------------------------------------------------
# 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;


# ------------------------------------------------------------------
# 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_USERN
AME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PAS
SWORD,
          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 */

  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_I
NT);
  OCIbbname(curr,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_I
NT);
  OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQLT_I
NT);
  OCIbbname(curr,l_pkey_bp,errhp,":l_pkey",ADR(l_pkey),SIZ(l_pkey),SQ
LT_INT);
  OCIbbname(curr,l_skey_bp,errhp,":l_skey",ADR(l_skey),SIZ(l_skey),SQ
LT_INT);
  OCIbbname(curr,l_quan_bp,errhp,":l_quan",ADR(l_quan),SIZ(l_quan),SQ
LT_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_F
LT);
  OCIbbname(curr,l_disc_bp,errhp,":l_disc",ADR(l_disc),SIZ(l_disc),SQ
LT_FLT);
  OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l_ep
rice),
            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_tp
rice),
            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_newepric
e),
            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_newtpric
e),
            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_DEF
AULT)) != 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=1"

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

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

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

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

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

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

#endif /* ATRANSPL_H */


# ------------------------------------------------------------
# ckpt.sh
# ------------------------------------------------------------

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

. $KIT_DIR/env

sqlplus -s /NOLOG << !

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

```
            exit;
!


# ----------------------------------------------------------------
# cnt_hist.sql
# ----------------------------------------------------------------

spool cnt_history
select count(*) from history;
spool off
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
#

. /home/oracle/kit/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=$KIT_DIR/utils/atranspl
USER=${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 $KIT_DIR/utils/randkey $ITER 1 u$USER
   $KIT_DIR/utils/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


# ----------------------------------------------------------------
# count_tx.sh
# ----------------------------------------------------------------

#!/bin/ksh

STEM=$1
ITER=$2
OUT=$3
FIN=FALSE
while [ "$FIN" = "FALSE" ]
do
  s=0
  FIN=TRUE
  while [ $s -lt $STEM ]
  do
    nt=`grep "Transaction Completed" $OUT/dura${s} | wc -l`
    if [ $nt -lt $ITER ];then
       FIN=FALSE
    fi
    s=`expr $s + 1`
  done
  sleep 5
done
echo all streams have commited $ITER transactions


# ----------------------------------------------------------------
# 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;


# ----------------------------------------------------------------
# dura.sh
# ----------------------------------------------------------------

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

. $KIT_DIR/env

# Create history table

# Count number of entries in the history table

SERVER="ultraperf2"

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

echo "------------------------------------------------------"
echo "Starting the durability tests `date`"
```

```
run_acid.sh &
echo "------------------------------------------------------"

sleep 1200

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

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

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

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

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

sleep 120

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


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

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

sleep 120

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

# Power Off


# ---------------------------------------------------------------
# end_acid.sh
# ---------------------------------------------------------------

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

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$OH/tpcd/audit set in env
OUT_DIR=$ACID_OUT/
DURA_DIR=$ACID_OUT/dura
RUN_ID_FILE=$ACID_DIR/run_id
SHELL_PID=`cat ${DURA_DIR}/shellpid`
ITER=100
STEM=${NUM_STREAMS}
let STEM="$STEM + 1" # add one for the update stream
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/dura
OUT=${DURA_DIR}/dura
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${SHELL_PID}_
USER=tpch/tpch
TRIG=1
HCNT=duracnta

# get history count

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

# perform the consistency

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

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


# ---------------------------------------------------------------
# gettime.c
# ---------------------------------------------------------------

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

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

/*

   NAME
     gettime.c

   DESCRIPTION
     get wall clock time.
     get cpu time.

   FUNCTIONS
     get wall clock time.
     get cpu time.

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

*/


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

#define SUN_OS5

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

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

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

```
#endif /* AIXRIOS */

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

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

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

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

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

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

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

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

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

# include <stdio.h>

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

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



double gettime ()

{

#ifdef TIME_W_GETTIME
   struct timeval tv;

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

#ifdef TIME_W_TIMES
   struct tms buf;

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

}



double getcpu ()

{

#ifdef CPU_W_TIMES
   struct tms buf;

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

#ifdef CPU_W_GETRU
   struct rusage ru;
   double usecs;

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

}



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

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

{

#ifdef GETRU_STATS
   struct rusage ru;

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

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

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

}



getru1 (kids)

int kids;

{

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

#ifdef GET_P_STATS
   timeval_t tv;

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

}



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

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

{

#ifdef GETRU_STATS
```

```
   struct rusage ru;

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

#ifdef GET_P_STATS
   timeval_t tv;
   struct process_stats ru;

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

}



#ifdef GETRU_STATS

print_ru (fp, ru)

FILE *fp;
struct rusage *ru;

{

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

}



diffru (ru2, ru)

struct rusage *ru2;
struct rusage *ru;

{

   ru2->ru_utime.tv_sec -= ru->ru_utime.tv_sec;
   ru2->ru_utime.tv_usec -= ru->ru_utime.tv_usec;
   ru2->ru_stime.tv_sec -= ru->ru_stime.tv_sec;
   ru2->ru_stime.tv_usec -= ru->ru_stime.tv_usec;
   ru2->ru_maxrss -= ru->ru_maxrss;
   ru2->ru_ixrss -= ru->ru_ixrss;
   ru2->ru_idrss -= ru->ru_idrss;
   ru2->ru_minflt -= ru->ru_minflt;
   ru2->ru_majflt -= ru->ru_majflt;
   ru2->ru_nswap -= ru->ru_nswap;
   ru2->ru_inblock -= ru->ru_inblock;
   ru2->ru_oublock -= ru->ru_oublock;
   ru2->ru_msgsnd -= ru->ru_msgsnd;
   ru2->ru_msgrcv -= ru->ru_msgrcv;
   ru2->ru_nsignals -= ru->ru_nsignals;
   ru2->ru_nvcsw -= ru->ru_nvcsw;
   ru2->ru_nivcsw -= ru->ru_nivcsw;

}

#endif /* GETRU_STATS */



#ifdef GET_P_STATS

print_ru (fp, ps)

FILE *fp;
struct process_stats *ps;

{

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

}



diffru (ru2, ru)

struct process_stats *ru2;
struct process_stats *ru;

{

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

}

#endif /* GET_P_STATS */


# ------------------------------------------------------------------
# gtime.c
# ------------------------------------------------------------------

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

/*

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

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

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

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

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

```
   NOTES
     <other useful comments, qualifications, etc.>

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

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

# include <sys/time.h>

main ()
{

  struct timeval tv;

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

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

}

/* end of file gtime.c */


# ------------------------------------------------------------------
# prepare4acid.sh
# ------------------------------------------------------------------

#!/bin/ksh
#
# $Header: prepare4acid.sh 12-aug-99.17:09:18 mpoess Exp $
#
# prepare4acid.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      prepare4acid.sh
#
#    DESCRIPTION
#     Prepares the qualification database for the acid tests.
#
#    NOTES
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/12/99 - Creation
#    mpoess      08/12/99 - Creation
#
. $KIT_DIR/env

sqlplus $DATABASE_USER @d_hist
sqlplus $DATABASE_USER @atrans


# ------------------------------------------------------------------
# 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_USERN
AME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PAS
SWORD,
          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_I
NT);
  OCIbbname(curi,o_key_bp,errhp,":o_key",ADR(o_key),SIZ(o_key),SQLT_I
NT);
}


# --------------------------------------------------------------------
# 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=$KIT_DIR/utils/atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=$DURA_DIR/dura
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
   $KIT_DIR/utils/randkey $ITER ${SF} u${USER} > ${KEY}${i} &
   i=`expr $i + 1`
done

wait
# perform the consistency

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

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

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

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

done

wait

echo "ACID run completed"


# --------------------------------------------------------------
# sample.sh
# --------------------------------------------------------------

#!/bin/ksh
#
# $Header: sample.sh 08-aug-99.17:10:00 mpoess Exp $
```

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

# $1 durability output file

. $KIT_DIR/env

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



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

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

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

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

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


# ------------------------------------------------------------
# sample.sql
# ------------------------------------------------------------

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

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

exit;
```

Appendix D: Query Text and Query Output

```
# ----------------------------------------------------------------
# 10.log
# ----------------------------------------------------------------

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


# ----------------------------------------------------------------
# 11.log
# ----------------------------------------------------------------

-- 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
154358.00              15212937.88
38823.00               15064802.86
---------------- rows deleted
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


# ----------------------------------------------------------------
```

```
# 12.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 13.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 14.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 15.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 16.log
# ------------------------------------------------------------

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
---------- rows deleted
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


# ----------------------------------------------------------------
# 17.log
# ----------------------------------------------------------------

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


# ----------------------------------------------------------------
# 18.log
# ----------------------------------------------------------------

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
Customer#000096103        96103.00               5984582.00
1992-03-16
494398.79              312.00
Customer#000024341        24341.00               1474818.00
1992-11-15
491348.26              302.00
Customer#000137446        137446.00              5489475.00
1997-05-23
487763.25              311.00
---------- rows deleted
Customer#000112987        112987.00              4439686.00
1996-09-17
418161.49              305.00
Customer#000012599        12599.00               4259524.00
1998-02-12
415200.61              304.00
Customer#000105410        105410.00              4478371.00
1996-03-05
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


# ----------------------------------------------------------------
# 19.log
# ----------------------------------------------------------------

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.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 20.log
# ------------------------------------------------------------

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
---------- rows deleted
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


# ------------------------------------------------------------
# 21.log
# ------------------------------------------------------------

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
--------- rows deleted
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


# ------------------------------------------------------------
# 22.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 2.log
# ------------------------------------------------------------

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
9923.77                Supplier#000002324        GERMANY
29821.00               Manufacturer#4
y3OD9UywSTOk                             17-779-299-1839
ackages boost blithely. blithely regular deposits c
9871.22                Supplier#000006373        GERMANY
43868.00               Manufacturer#5
J8fcXWsTqM                               17-813-485-8637
etect blithely bold asymptotes. fluffily ironic platelets wake
furiously; blit
9870.78                Supplier#000001286        GERMANY
81285.00               Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH    17-516-924-4574
 regular accounts. furiously unusual courts above the fi
9870.78                Supplier#000001286        GERMANY
181285.00              Manufacturer#4
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH    17-516-924-4574
 regular accounts. furiously unusual courts above the fi
9852.52                Supplier#000008973        RUSSIA
18972.00               Manufacturer#2
t5L67YdBYYH6o,Vz24jpDyQ9                 32-188-594-7038
rns wake final foxes. carefully unusual depende
9847.83                Supplier#000008097        RUSSIA
130557.00              Manufacturer#2
xMe97bpE69NzdwLoX                        32-375-640-3593
 the special excuses. silent sentiments serve carefully final ac
9847.57                Supplier#000006345        FRANCE
86344.00               Manufacturer#1
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag   16-886-766-7945
ges. slyly regular requests are. ruthless, express excuses cajole
blithely across the unu
--------------------  rows deleted
33094.00               Manufacturer#5
kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj        17-627-663-8014
arefully unusual requests x-ray above the quickly final deposits.
7937.93                Supplier#000009012        ROMANIA
83995.00               Manufacturer#2
iUiTziH,Ek3i4lwSgunXMgrcTzwdb            29-250-925-9690
to the blithely ironic deposits nag sly
7914.45                Supplier#000001013        RUSSIA
125988.00              Manufacturer#2
riRcntps4KEDtYScjpMIWeYF6mNnR            32-194-698-3365
 busily bold packages are dolphi
7912.91                Supplier#000004211        GERMANY
159180.00              Manufacturer#5
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG        17-266-947-7315
ay furiously regular platelets. cou
7912.91                Supplier#000004211        GERMANY
184210.00              Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG        17-266-947-7315
ay furiously regular platelets. cou
7894.56                Supplier#000007981        GERMANY
85472.00               Manufacturer#4
NSJ96vMROAbeXP                           17-963-404-3760
ic platelets affix after the furiously
7887.08                Supplier#000009792        GERMANY
164759.00              Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz      17-988-938-4296
ckly around the carefully fluffy theodolites. slyly ironic pack
7871.50                Supplier#000007206        RUSSIA
104695.00              Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW                  32-432-452-7731
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


# ------------------------------------------------------------
# 3.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 4.log
# ------------------------------------------------------------

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.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 6.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 7.log
# ------------------------------------------------------------

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
```

```
# ------------------------------------------------------------
# 8.log
# ------------------------------------------------------------

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


# ------------------------------------------------------------
# 9.log
# ------------------------------------------------------------

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
-------------------------- rows deleted
UNITED STATES           1994.00                 49296747.18
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
```

Appendix E: Seed and Input Parameters

```
# ------------------------------------------------------------
# qp1.0
# ------------------------------------------------------------

14        1995-04-01
2             40    BRASS     AMERICA
9          goldenrod
20           deep   1997-01-01          ETHIOPIA
6         1997-01-01          0.04      25
17         Brand#52 MED BAG
18            313
8           KENYA  AFRICA    MEDIUM ANODIZED BRASS
21          RUSSIA
13          express requests
3          FURNITURE1995-03-09
22            15    16        27        31        20        25
              33
16         Brand#15 SMALL BRUSHED       23        50        45
              29    5         17        15        43
4         1994-06-01
11           IRAN   0.0000001000
15        1996-09-01
1             110
10        1993-07-01
19         Brand#32 Brand#52  Brand#25  7         14        21
5         MIDDLE EAST         1997-01-01
7           GERMANY KENYA
12          REG AIR SHIP      1995-01-01


# ------------------------------------------------------------
# qp1.1
# ------------------------------------------------------------

21          KENYA
3          MACHINERY1995-03-26
18            314
5           AFRICA  1997-01-01
11       UNITED KINGDOM        0.0000001000
7        UNITED STATES         FRANCE
6          1997-01-01          0.09      24
20            pale   1995-01-01          SAUDI ARABIA
17         Brand#54 MED PACK
12           SHIP   TRUCK     1994-01-01
16         Brand#45 ECONOMY BURNISHED   27        33        14
              49    22        44        21        28
15        1994-06-01
13          express requests
10        1994-04-01
2             28    TIN       MIDDLE EAST
8           FRANCE  EUROPE    SMALL POLISHED BRASS
14        1995-07-01
19         Brand#34 Brand#35  Brand#14  2         15        28
9          firebrick
22            30    17        21        34        19        10
              11
1             118
4         1997-01-01


# ------------------------------------------------------------
# qp1.2
# ------------------------------------------------------------

6         1997-01-01          0.07      25
17         Brand#51 MED DRUM
14        1995-11-01
16         Brand#35 STANDARD PLATED     30        46        15
              21    20        38        35        49
19         Brand#31 Brand#23  Brand#13  7         16        25
10        1995-01-01
9             cyan
2             15    COPPER    AMERICA
15        1996-12-01
8        UNITED KINGDOM        EUROPE    SMALL BURNISHED STEEL
5           AMERICA 1997-01-01
22            20    18        26        27        10        21
              11
12            FOB   REG AIR   1995-01-01
7         MOZAMBIQUE          UNITED KINGDOM
13          special accounts
18            312
1              65
4         1994-10-01
20           black  1994-01-01          IRAN
3          FURNITURE1995-03-12
11           IRAQ   0.0000001000
21          FRANCE


# ------------------------------------------------------------
# qp1.3
# ------------------------------------------------------------

8           MOROCCO AFRICA    STANDARD BRUSHED STEEL
5             ASIA  1997-01-01
4         1997-05-01
6         1997-01-01          0.04      25
17         Brand#53 JUMBO BAG
7            INDIA  MOROCCO
1              73
18            313
22             34    11        22        13        19        15
               18
14        1996-02-01
9           chiffon
10        1993-10-01
15        1994-09-01
11       UNITED STATES         0.0000001000
20           lime   1997-01-01          UNITED STATES
2               3    STEEL     MIDDLE EAST
21       UNITED KINGDOM
19          Brand#43 Brand#55  Brand#13  3         17        21
13          special accounts
16          Brand#15 MEDIUM BRUSHED      33        46        41
               50    45        22        10        35
12            MAIL   REG AIR   1996-01-01
3          MACHINERY1995-03-28


# ------------------------------------------------------------
# qp1.4
# ------------------------------------------------------------

5            EUROPE  1993-01-01
21          MOROCCO
14        1996-05-01
19         Brand#45 Brand#43  Brand#52  8         18        28
15        1997-04-01
17         Brand#55 JUMBO PKG
12           RAIL   REG AIR   1996-01-01
6         1993-01-01          0.02      24
4         1995-02-01
9             blue
8           GERMANY EUROPE    STANDARD PLATED STEEL
16         Brand#45 PROMO ANODIZED      2         28        32
              14    8         6         1         43
11           JAPAN  0.0000001000
2               41    BRASS     ASIA
10        1994-08-01
18            315
1              81
13          special accounts
7           ALGERIA GERMANY
22            11    26        25        31        12        14
              15
3          BUILDING1995-03-14
20           steel  1996-01-01          KENYA


# ------------------------------------------------------------
# qp1.5
# ------------------------------------------------------------

21          GERMANY
15        1995-01-01
4         1997-09-01
6         1993-01-01          0.07      25
7              PERU  UNITED STATES
16          Brand#35 SMALL PLATED        1         4         16
              20    27        36        28        45
19         Brand#42 Brand#21  Brand#51  3         19        24
18            312
14        1996-08-01
22            30    26        13        24        22        25
              11
11          ALGERIA 0.0000001000
13          special accounts
3          MACHINERY1995-03-30
1              89
2              29    NICKEL    MIDDLE EAST
5           AFRICA  1993-01-01
8        UNITED STATES         AMERICA   STANDARD ANODIZED STEEL
20         gainsboro1994-01-01          EGYPT
12            AIR   REG AIR   1996-01-01
17         Brand#52 JUMBO DRUM
10        1993-05-01
9         aquamarine


# ------------------------------------------------------------
# qp1.6
# ------------------------------------------------------------

10        1994-02-01
3          BUILDING1995-03-16
15        1997-07-01
13          special deposits
6         1993-01-01          0.04      25
8         MOZAMBIQUE          AFRICA    PROMO POLISHED STEEL
```

```
9           violet
7          INDONESIAMOZAMBIQUE
4         1995-06-01
11          JORDAN  0.0000001000
22            29    26        15        10        12        20
              25
18            314
12          REG AIR AIR       1996-01-01
1             97
5           AMERICA 1993-01-01
16          Brand#15 LARGE POLISHED      3         38        26
                2    9         49        40        18
2             17    COPPER    ASIA
14        1996-11-01
19          Brand#54 Brand#14  Brand#55  8         20        20
20           purple  1993-01-01          ROMANIA
17          Brand#54 WRAP BAG
21          ALGERIA


# ----------------------------------------------------------------
# qp1.7
# ----------------------------------------------------------------

18            312
8            INDIA  ASIA      PROMO BURNISHED COPPER
20          chiffon 1996-01-01          INDIA
21            PERU
2              4    STEEL     AFRICA
4         1993-03-01
22            16    20        19        10        29        28
              17
17          Brand#51 WRAP PKG
1             105
11         ARGENTINA0.0000001000
9            spring
19          Brand#51 Brand#42  Brand#45  4         10        28
3          HOUSEHOLD1995-03-01
13          special deposits
5             ASIA   1993-01-01
7           ALGERIA INDIA
10        1994-11-01
16          Brand#45 STANDARD ANODIZED   47        5         20
```

```
              8    4         29        17        18
6         1993-01-01          0.02      24
14        1997-03-01
15        1995-04-01
12           SHIP   AIR       1997-01-01


# ----------------------------------------------------------------
# qp1.8
# ----------------------------------------------------------------

19          Brand#53 Brand#25  Brand#44  9         11        24
1              113
15        1997-11-01
17          Brand#52 WRAP DRUM
5            EUROPE  1994-01-01
8           ALGERIA AFRICA    ECONOMY BRUSHED COPPER
9           seashell
12             FOB   AIR       1997-01-01
14        1997-06-01
7              PERU   ALGERIA
4         1995-10-01
3         AUTOMOBILE          1995-03-18
20           misty  1994-01-01          UNITED KINGDOM
16          Brand#35 MEDIUM BURNISHED    17        14        46
               48    36        34        3         31
6         1994-01-01          0.07      24
22             11    27        26        13        32        19
               29
10        1993-08-01
13           special deposits
2              42    BRASS     ASIA
21          INDONESIA
18             313
11            KENYA  0.0000001000


# ----------------------------------------------------------------
# seed
# ----------------------------------------------------------------

1019000532
```

Appendix F: Benchmark Programs and Scripts

```
# ------------------------------------------------------------
# dbinsert.sql
# ------------------------------------------------------------

rem
rem
================================================================+
rem  FILENAME
rem       inserts.sql
rem  DESCRIPTION
rem       Inserts duplicate rows with new key numbers and
rem       inserts rows with values beyond the TPC-D values.
rem
rem
================================================================
rem
rem Usage:   sqlplus tpcc/tpcc @insert
rem


set pagesize 100
set termout on
set echo on
set timing on
spool rdbinsert


rem
================================================================
rem Duplicates
rem
================================================================

REM get timestamp
select to_char (sysdate, 'HH24:MI:SS') timestamp from dual;


drop table temp_part;
create table temp_part as
   select * from part
     where p_partkey = 1;
update temp_part
  set p_partkey = 2147483647;
insert into part
  (select * from temp_part);
select * from part
 where p_partkey = 2147483647
       or p_partkey = 1;
delete from part
 where p_partkey = 2147483647;
drop table temp_part;
commit;

drop table temp_supplier;
create table temp_supplier as
   select * from supplier
     where s_suppkey = 1;
update temp_supplier
  set s_suppkey = 2147483647;
insert into supplier
(select * from temp_supplier);
select * from supplier
 where s_suppkey = 2147483647
       or s_suppkey = 1;
delete from supplier
 where s_suppkey = 2147483647;
drop table temp_supplier;
commit;

drop table temp_partsupp;
create table temp_partsupp as
   select * from partsupp
     where ps_partkey = 1
      and ps_suppkey = 2;
update temp_partsupp
 set ps_partkey = 2147483647,
     ps_suppkey = 2147483647;
insert into partsupp
(select * from temp_partsupp);
select * from partsupp
where (ps_partkey = 2147483647
and    ps_suppkey = 2147483647)
     or    (ps_partkey = 1
    and    ps_suppkey = 2);
delete from partsupp
 where ps_partkey = 2147483647
and   ps_suppkey = 2147483647;
drop table temp_partsupp;
commit;


drop table temp_customer;
create table temp_customer as
   select * from customer
     where c_custkey = 1;
update temp_customer
  set c_custkey = 2147483647;
insert into customer
(select * from temp_customer);
select * from customer
 where c_custkey = 2147483647
       or c_custkey = 1;
delete from customer
 where c_custkey = 2147483647;
drop table temp_customer;
commit;

drop table temp_orders;
create table temp_orders as
   select * from orders
where o_orderkey = (select min(o_orderkey) from orders);
update temp_orders
 set o_orderkey = 2147483647;
insert into orders
 (select * from temp_orders);
select * from orders
 where o_orderkey = 2147483647
or o_orderkey = (select min(o_orderkey) from orders);
delete from orders
where o_orderkey = 2147483647;
drop table temp_orders;
commit;

drop table temp_lineitem;
create table temp_lineitem as
   select * from lineitem
where l_orderkey = (select min(o_orderkey) from orders)
    and   l_linenumber = 1;
update temp_lineitem
 set l_orderkey = 2147483647,
     l_partkey  = 2147483647,
     l_suppkey  = 2147483647,
    l_linenumber = -2147483646;
insert into lineitem
(select * from temp_lineitem);
select * from lineitem
where (l_orderkey = 2147483647
and    l_partkey  = 2147483647
and    l_suppkey  = 2147483647
and    l_linenumber = -2147483646)
          or    (l_orderkey = (select min(o_orderkey) from orders)
   and    l_linenumber = 1);
delete from lineitem
 where l_orderkey = 2147483647
 and   l_partkey  = 2147483647
 and   l_suppkey  = 2147483647
and   l_linenumber = -2147483646;
drop table temp_lineitem;
commit;

drop table temp_nation;
create table temp_nation as
   select * from nation
    where n_nationkey = 1;
update temp_nation
 set n_nationkey = 2147483647;
insert into nation
 (select * from temp_nation);
select * from nation
where n_nationkey = 2147483647
      or n_nationkey = 1;
delete from nation
where n_nationkey = 2147483647;
drop table temp_nation;
commit;


drop table temp_region;
create table temp_region as
   select * from region
    where r_regionkey = 1;
update temp_region
 set r_regionkey = 2147483647;
insert into region
 (select * from temp_region);
select * from region
where r_regionkey = 2147483647
      or r_regionkey = 1;
delete from region
where r_regionkey = 2147483647;
drop table temp_region;
commit;

rem
================================================================
rem Duplicates finished starting inserts for domain range
rem
================================================================

REM get timestamp
```

```
select to_char (sysdate, 'HH24:MI:SS') timestamp from dual;


insert into supplier
(s_suppkey, s_name, s_address, s_nationkey, s_phone,
    s_acctbal, s_comment)
           values
(2147483647, 'NAME text ............25E',
'Address varchar ............30.......40E',
2147483647,'This is phone E', 123456789012,
'Supplier comment field is 101 long no E');
select * from supplier
 where s_suppkey = 2147483647;
delete from supplier D
 where s_suppkey = 2147483647;
rem
================================================================
insert into part
(p_partkey, p_name, p_mfgr, p_brand, p_type,
p_size, p_container, p_retailprice, p_comment)
           values
(2147483647, 'Pname text .......2.........3.........4....5E',
'Pmfgr text........2....5E','Pbrand 10E',
'Ptype varchar.....2....5E', 2147483646,
  'PcontainrE', 123456789012,
  'Part comment field  23E');
select * from part
 where p_partkey = 2147483647;
delete from part
 where p_partkey = 2147483647;
rem
================================================================
insert into partsupp
(ps_partkey, ps_suppkey, ps_availqty, ps_supplycost,
         ps_comment)
           values
(2147483647, 2147483647, -2147483646, 123456789012,
'PS comment field is 199 long no E');
select * from partsupp
 where ps_partkey = 2147483647
and   ps_suppkey = 2147483647;
delete from partsupp
 where ps_partkey = 2147483647
and   ps_suppkey = 2147483647;
rem
================================================================
insert into customer
(c_custkey, c_name, c_address, c_nationkey,
c_phone, c_acctbal, c_mktsegment, c_comment)
           values
(2147483647, 'Customer Name goes to 25E',
'Customer Address goes here..3.........4E',
2147483647, 'This is phone E', 123456789012,
'ZMark segE', 'Customer comments fiels is 117 long no E');
select * from customer
 where c_custkey = 2147483647;
delete from customer
 where c_custkey = 2147483647;
rem
================================================================
insert into orders
(o_orderkey, o_custkey, o_orderstatus, o_totalprice,
o_orderdate, o_orderpriority, o_clerk, o_shippriority,
          o_comment)
           values
(2147483647, 2147483647, 'X', 123456789012,
TO_DATE('2005-12-30','YYYY-MM-DD'),
'Order Priorty5E', 'Fixed text  15E', -2147483646,
'Order comments field is 79 no E');
select * from orders
 where o_orderkey = 2147483647
 and  o_custkey = 2147483647;
delete from orders
 where o_orderkey = 2147483647
 and  o_custkey = 2147483647;
rem
================================================================
insert into lineitem
(l_orderkey, l_partkey, l_suppkey, l_linenumber,
l_quantity, l_extendedprice, l_discount, l_tax,
l_returnflag, l_linestatus, l_shipdate, l_commitdate,
l_receiptdate, l_shipinstruct, l_shipmode, l_comment)
           values
         (2147483647,
          2147483647,
          2147483647,
         -2147483646,
        -123456789012,
        -123456789012,
        -123456789012,
        -123456789012,
             'Q',
             'R',
TO_DATE('2005-12-30','YYYY-MM-DD'),
TO_DATE('2005-12-30','YYYY-MM-DD'),
TO_DATE('2005-12-30','YYYY-MM-DD'),
 'Ship by camel .........5E',
        'Ship ASAPE',
'Is this really what you wanted? 44 long....E');
select * from lineitem
 where l_orderkey = 2147483647
  and l_partkey = 2147483647
  and l_suppkey = 2147483647
and l_linenumber = -2147483646;
delete from lineitem
 where l_orderkey = 2147483647
  and l_partkey = 2147483647
  and l_suppkey = 2147483647
and l_linenumber = -2147483646;
rem
================================================================
insert into nation
(n_nationkey, n_name, n_regionkey, n_comment)
           values
        (2147483647,
 'Ze Republic d MakebelievE',
         2147483647,
'A nation comment for field size 152 no E');
select * from nation
where n_nationkey = 2147483647
 and n_regionkey = 2147483647;
delete from nation
where n_nationkey = 2147483647
 and n_regionkey = 2147483647;
rem
================================================================
insert into region
(r_regionkey, r_name, r_comment)
           values
        (2147483647,
 'Ze ends of the earth....E',
'A reasonable comment would go herE');
select * from region
where r_regionkey = 2147483647;
delete from region
where r_regionkey = 2147483647;
rem
================================================================

REM get timestamp
select to_char (sysdate, 'HH24:MI:SS') timestamp from dual;
rem
================================================================
rem Done
rem
================================================================
spool off;
exit;


# ----------------------------------------------------------------
# dbtables.sql
# ----------------------------------------------------------------

set echo on
set numwidth 25
spool rdbtablest
SELECT COUNT(*) FROM LINEITEM;

SELECT * FROM LINEITEM
WHERE L_ORDERKEY IN
 ( 4, 26598,  148577,  387431, 56704, 517442, 600000)
AND L_LINENUMBER = 1
ORDER BY L_ORDERKEY;



SELECT * FROM REGION;

SELECT COUNT(*) FROM NATION;

SELECT * FROM NATION
WHERE N_NATIONKEY IN (3,10,14,20)
ORDER BY N_NATIONKEY;



SELECT COUNT(*) FROM ORDERS;

SELECT * FROM ORDERS
WHERE O_ORDERKEY IN ( 7, 44065, 287590, 411111, 483876, 599942 )
ORDER BY O_ORDERKEY;



SELECT COUNT(*) FROM PART;

SELECT * FROM PART
WHERE P_PARTKEY IN (1,984,8743,9028,13876,17899,20000)
ORDER BY P_PARTKEY;



SELECT COUNT(*) FROM PARTSUPP;

SELECT* FROM PARTSUPP
```

```
   WHERE PS_PARTKEY = 3398
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 3398);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY =15873
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 15873);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 11394
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 11394);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 6743
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY = 6743);

SELECT* FROM PARTSUPP
   WHERE PS_PARTKEY = 19763
   AND PS_SUPPKEY = (SELECT MIN(PS_SUPPKEY)
      FROM PARTSUPP WHERE PS_PARTKEY =19763);



SELECT COUNT(*) FROM SUPPLIER;

SELECT * FROM SUPPLIER
WHERE  S_SUPPKEY IN (83,265,492,784,901,1000)
ORDER BY S_SUPPKEY;

SELECT COUNT(*) FROM CUSTOMER;

DROP TABLE MINMAX;

CREATE TABLE MINMAX
(TNAME CHAR(15),
KEYMIN INTEGER,
KEYMAX INTEGER);

INSERT INTO MINMAX
SELECT 'LINEITEM_ORD',MIN(L_ORDERKEY),MAX(L_ORDERKEY)
FROM LINEITEM ;

INSERT INTO MINMAX
SELECT 'LINEITEM_NBR',MIN(L_LINENUMBER),MAX(L_LINENUMBER)
FROM LINEITEM;

INSERT INTO MINMAX
SELECT 'ORDERTBL',MIN(O_ORDERKEY),MAX(O_ORDERKEY)
FROM ORDERS;

INSERT INTO MINMAX
SELECT 'CUSTOMER',MIN(C_CUSTKEY),MAX(C_CUSTKEY)
FROM CUSTOMER;

INSERT INTO MINMAX
SELECT 'PART',MIN(P_PARTKEY),MAX(P_PARTKEY)
FROM PART;

INSERT INTO MINMAX
SELECT 'SUPPLIER',MIN(S_SUPPKEY),MAX(S_SUPPKEY)
FROM SUPPLIER;

INSERT INTO MINMAX
SELECT 'PARTSUPP_PART',MIN(PS_PARTKEY),MAX(PS_PARTKEY)
FROM PARTSUPP;

INSERT INTO MINMAX
SELECT 'PARTSUPP_SUPP',MIN(PS_SUPPKEY),MAX(PS_SUPPKEY)
FROM PARTSUPP ;

INSERT INTO MINMAX
SELECT 'NATION',MIN(N_NATIONKEY),MAX(N_NATIONKEY)
FROM NATION;

INSERT INTO MINMAX
SELECT 'REGION',MIN(R_REGIONKEY),MAX(R_REGIONKEY)
FROM REGION;

SELECT * FROM MINMAX;
spool off
exit;


# ----------------------------------------------------------------
# firstten.sql
# ----------------------------------------------------------------

set echo on
set numwidth 25
spool count.out
select * from lineitem where rownum < 11;
select * from orders where rownum < 11;
select * from part where rownum < 11;
select * from partsupp where rownum < 11;
select * from supplier where rownum < 11;
select * from customer where rownum < 11;
select * from nation where rownum < 11;
select * from region where rownum < 11;
spool off
exit;


# ----------------------------------------------------------------
# gen_seed.sh
# ----------------------------------------------------------------

#!/bin/ksh

SEED_FILE=$1

#Generate the seed
echo "Setting the random number seed"
PSEED=`date +%m:%d:%H:%M:%S | sed -e 's/://g'`
echo "Using ${PSEED} as seed0"
echo ${PSEED} >  $SEED_FILE
echo "Done setting the random number seed"


# ----------------------------------------------------------------
# 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 */


# ----------------------------------------------------------------
# qexecpl.c
# ----------------------------------------------------------------

#ifdef RCSID
static char *RCSid =
   "$Header: qexecpl.c 17-oct-2001.09:29:47 mpoess Exp $ ";
#endif /* RCSID */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights
reserved.  */

/*

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

   DESCRIPTION
     SQL Execution Engine, Oracle v8,  OCI version

   PRIVATE FUNCTION(S)
     <list of static functions defined in .c file - with one-line
```

```
descriptions>

   MODIFIED   (MM/DD/YY)
      mpoess    10/17/01 - add serialization level in SQLinit
      mpoess    02/22/01 - add linux changes
      mpoess    08/05/99 - make compile
      mpoess    11/13/98 - fix pddl statement
      pswong    02/19/97 - migrating to version 8
      pswong    04/02/96 - more polishing
      pswong    03/25/96 - polish up
      pswong    03/06/96 - created

*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

int define_output_variables();
void process_select_list();
void usage();
void SQLinit();
void SQLexec();
void SQLexit();
void *memalloc();
void print_header();
void print_rows();
int OFEN();
void remove_newline();

char logname[UNAME_LEN];  /* username/passwd combo */
char *passwd;

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

double s_tr_start = 0.0;  /* statement start time                */
double s_tr_end = 0.0;    /* statement end time                  */

/* For our purpose of timing, we will treat comments as delimiters */
/* for queries.   Thus, we will collect query timings whenever we  */
/* encounter a comment (of course not for the first comment in a   */
/* file).                                                          */

int end_flag = 0;         /* flag to indicate that we have reached */
                          /* the end of a query                    */

int stmt_cnt = 0;         /* Number of statements processed.     */
int qry_cnt = 0;          /* Number of query processed.          */

double product = 1.0;     /* cumulative product of query times   */
int rows_ret = 0;         /* the number of rows fetched          */
int num_sel_list = 0;     /* the number of select list item      */

long num_to_fetch = -1;   /* Number of rows to fetch. -1 means fetch
all */

sltype slist[MAX_SEL_LIST];  /* Array for describing Select List
*/
dltype *dlist[MAX_SEL_LIST]; /* Array of ptrs for Defining Select
List   */

char stmt[SQL_LEN];       /* The SQL statement or comment line.  */
char qn[4];               /* Number of the query being executed  */
char qnp[4];              /* Number of the previous query executed
*/
char cmnt[5000];            /* Buffer to save the comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template               */
FILE *logfile;   /* log and report files                */
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template               */
FILE *logfile = stdout;   /* log and report files                */
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN           */
int deflen = 0;           /* Size of data type for ODEFIN        */
int deftype = 1;          /* Oracle type number for ODEFIN       */

int pfmem = PFMEMSIZE;    /* Memory to prefetch rows             */

time_t tim;               /* To get wall clock time              */

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curq = NULL;
OCIStmt *cur_dml = NULL;
OCIStmt *cur_ddl = NULL;
OCIParam *tpcpar = NULL;

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

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

void usage() {

  fprintf(stderr,"\nUsage: qexec username/password [q<path name for
query template file>]\n");
  fprintf(stderr,"                 [l<path name for log>] [r<path
name for reports>]\n\n");
  fprintf(stderr,"Options:\n");
  fprintf(stderr,"q<path for query>         : full path name for the
query template file.\n");
  fprintf(stderr,"                            (default is stdin)\n");
  fprintf(stderr,"l<path name for log>      : full path name for log
files\n");
  fprintf(stderr,"                            (default is
stdout)\n");
  fprintf(stderr,"r<path name for reports>  : full path name for
reports\n");
  fprintf(stderr,"                            (default is
stdout)\n");
  exit(-1);
}



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

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

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

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}

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

  int i,pos,pos2;
  int retcode;      /* Return code for get_statement */
#ifdef LINUX
  logfile=fopen("/dev/stdout","w");
  qtemp=fopen("/dev/stdin","rw");
  rep=fopen("/dev/stdout","w");
#endif
  /* Initialize some variables */

  if ((argc > 5) || (argc < 2)) {
    usage();
  }

  /* argv[1] -- User and Password for Database */

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'q':
      if ((qtemp = fopen(++(argv[0]),"r")) == NULL) {
fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
          exit(-1);
      }
      break;
    case 'r':
      if ((rep = fopen(++(argv[0]),"a")) == NULL) {
fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
          exit(-1);
      }
      break;
    case 'l':
      if ((logfile = fopen(++(argv[0]),"a")) == NULL) {
fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
          exit(-1);
      }
      break;
    default:
      fprintf(stderr,"Invalid Option: %c\n", argv[0][0]);
      usage();
      break;
    }
  }

  /* Do some initialization and establish connection with the
database */

  SQLinit();

  /* May want to add some triggering mechanism here */

  time(&tim);
  fprintf(logfile, "Begin Execution at %s\n\n", ctime(&tim));
  fprintf(rep, "Begin Executing this Stream at %s\n\n", ctime(&tim));
  /* Get the next statement and start processing it */

  while ((retcode = get_statement()) > 0) {

    switch (retcode) {

      /* If this is a comment, skips it */
    case COMMENT:
      /*if (end_flag) {
end_flag = 0;  */    /* reset query end flag */
/* save the comment so that we can print it out later on  */
    /* strcpy(cmnt, stmt);
          break;
      } */
      if (stmt[3]== '@') {
         pos=4;
         strcpy(qnp,qn);
         while (stmt[pos] != ')') {
           pos++;
         }
         pos2=0;
         pos++;
         while (stmt[pos] != '.') {
           /*printf ("qn %d %c \n",pos2,stmt[pos]);*/
           qn[pos2]=stmt[pos];
           pos2++;
           pos++;
         }
         qn[pos2] = 0;
         /* printf("found a new query: %s\n",qn); */
      }
      /* save the comment so that we can print it out later on  */
      strcat(cmnt, stmt);
      break;

      /* if this is a set_row_fetch command */
    case SET_FETCHROW:
      fprintf(logfile,"Setting the number of rows to fetch to:
%ld\n\n",
           num_to_fetch);
      break;

      /* if this is a SQL statement */
    case SQL_STMT:

      /* Executes the query */
      SQLexec();

      stmt_cnt++;
      qry_cnt++;
      fflush(rep);
      fflush(logfile);
      /*
      fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
      fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);

      fprintf(logfile,"Statement Processed in %.2f seconds.\n",
      (s_tr_end - s_tr_start));
      fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
                     qn,(s_tr_end - s_tr_start)s_tr_start,s_tr_end);
      fflush(rep);
      fflush(logfile);*/
      break;

      /* Should never reach here */
    default:
      fprintf(stderr, "Invalid statement type!!\n");
      SQLexit();
      break;
    }
  }

  /* Get Timing for the last query */

  tr_end = gettime();

  fprintf(logfile,"Query Processed in %.2f seconds.\n\n",(tr_end -
s_tr_start));

  /* print comments for this query that we have saved */

  /* fprintf(logfile, "%s\n", cmnt); */

  /* fprintf(rep, "Query %s : Execution time %.2f\n", qn,(tr_end -
s_tr_start));*/
  fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
                 qn,(tr_end - s_tr_start),s_tr_start,tr_end);

  time(&tim);
  fprintf(logfile,"\nEnded Executing this Stream at %s\n",
ctime(&tim));
  fprintf(logfile,"\nStream Started at %.2f\n", tr_start);
  fprintf(logfile,"Stream Ended at %.2f\n", tr_end);
  fprintf(logfile,"Stream Processed in %.2f seconds\n\n",(tr_end -
tr_start));

  fprintf(rep,"\nEnded Executing this Stream at %s\n", ctime(&tim));
  fprintf(rep,"\nStream Started at %.2f\n", tr_start);
  fprintf(rep,"Stream Ended at %.2f\n", tr_end);
  fprintf(rep,"Stream Processed in %.2f seconds\n\n",
      (tr_end - tr_start));

  fprintf(logfile, "\nSQL statements processed: %d\n", stmt_cnt);
  /*fprintf(logfile, "Queries processed: %d\n", qry_cnt);*/

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */
```

```
  SQLexit();
  exit(0);
}


/* SQLinit(): Perform initialization tasks.
*/
/*            Logs on to Oracle, opens some files and open a cursor
for */
/*            later use.
*/

void SQLinit() {

  int i;

  /* preallocate MAX_PREALLOC members of the dlist array
*/
  /* initializes others to NULL so that we can determine who to free
later */

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc (sizeof(dltype));
      dlist[i]->defhdl = NULL;
/*      OCIhalloc(curq,&(dlist[i]->defhdl),OCI_HTYPE_DEFINE); */
    }
    else
      dlist[i] = NULL;
  }

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

  (void) OCIInitialize(OCI_DEFAULT,(dvoid *)0,0,0,0);

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

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

  /* get username and password */

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

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

  OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SERVER,errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(logname),OCI_ATTR_U
SERNAME,
          errhp);
  OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI_ATTR_PAS
SWORD,
          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);

/*
  if ((status=OCILogon((OCIEnv *)tpcenv,(OCIError *)errhp,(OCISvcCtx
*)tpcsvc,
                       (text *)logname, strlen(logname), (text
*)passwd,
                       strlen(passwd), (text *) 0, 0)) != OCI_SUCCESS)
    sql_error(errhp, status, 1);
*/
  printf("\nConnected to ORACLE as user: %s\n\n", logname);

}



/* SQLexec()  Executes the SQL statement.
*/
/*            Parse the SQL statement.
*/
/*            If DDL or DML statements, execute right away.
*/
/*            Else describe and define select list outputs,
*/
/*                 execute and fetch results.
*/

void SQLexec()
{
  int i;
  ub2 stmttyp = OCI_STMT_SELECT;     /* default is a SELECT statement
*/

  /* Clause 5.3.6.2: QI(i,s) is the time between the first character
*/
  /*                 of this query text is submitted and the first
*/
  /*                 character of the next query text is submitted.
*/

  if (qry_cnt) {
     time(&tim);
     s_tr_end = gettime();
     fprintf(logfile,"Query Processed in %.2f seconds.\n\n",
      (s_tr_end - s_tr_start));

     /* print comments for this query that we have saved */

     /* fprintf(logfile, "%s\n", cmnt); */

     /*fprintf(rep, "Query %s : Execution time %.2f\n", qnp,(s_tr_end
- s_tr_start));*/
     fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
                    qnp,(s_tr_end - s_tr_start),s_tr_start,s_tr_end);

     /* Let's fflush stuff so that we can see what's going on */

     /* Fix for Q15 */
     fflush(logfile);
     fflush(rep);

     strcpy(qnp,qn);

     fflush(logfile);
     fflush(rep);
  }
  else
     tr_start = gettime();

  s_tr_start = gettime();

  /* prepare the statement */

  if ((status = OCIStmtPrepare(curq, errhp, (text*) stmt, (ub4)
strlen(stmt),
                               OCI_NTV_SYNTAX, OCI_DEFAULT)) !=
OCI_SUCCESS)
     sql_error(errhp,status,1);

  /* Prints the query text and comment to the logfile */

  fprintf(logfile, "\n%s\n", cmnt);
  cmnt[0]=0;
  fprintf(logfile, "\n%s\n", stmt);

  /* if this is a DDL or DML statement, execute it right away */
  /* only worries about SELECT statements right now, cannot   */
  /* execute a stored PL/SQL procedure in thie version        */

  OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_STMT_TYPE,errhp)
;

  if (stmttyp != OCI_STMT_SELECT) {
    OCIsexec(tpcsvc,curq,errhp,1);
    return;
  }

  /* otherwise, this is a select statement */
  /* Describe and define output variables  */

  /* first let's execute it to get the select-list definition */

  OCIaset(curq, OCI_HTYPE_STMT, &pfmem, 0, OCI_ATTR_PREFETCH_MEMORY,
errhp);

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

  /* Need to get the number of rows fetched first   */
  /* since the following statments will screw it up */

  OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW_COUNT,errhp
);

  /* To control memory usage, let's free up the extra dlist entries
*/
  /* that we have allocated.
```

```
*/

  i=MAX_PREALLOC;
  while(dlist[i] != NULL) {
    free(dlist[i]);
    dlist[i++] = NULL;
  }

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

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

  /* free all memory */

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (dlist[i] != NULL) {
      free(dlist[i]);
    }
  }

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


/* define_output_variables(): Describe and define select-list items
for */
/*                            a query statement.
*/
/*                            Returns the number of select-list items
*/
/*                            for this query.
*/

int define_output_variables()
{

  int i;
  int retflag = 0;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

    if (OCIParamGet(curq, OCI_HTYPE_STMT, errhp, (dvoid **) &tpcpar,
                                    POS(i)) != OCI_SUCCESS)
      break;


    /* dsize and nullok fields of dlist not used */

    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbsize),
    NULL, OCI_ATTR_DATA_SIZE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbtype),
    NULL, OCI_ATTR_DATA_TYPE, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].buf),
    &(slist[i].buflen), OCI_ATTR_NAME, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].precision),
    NULL, OCI_ATTR_PRECISION, errhp);
    OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].scale),
    NULL, OCI_ATTR_SCALE, errhp);

    /* For formatting purpose, remove trailing blanks in select-list
name. */

/*
    if (slist[i].buflen < MAX_COLNAME_SIZE)
      (slist[i].buf)[slist[i].buflen] = '\0';
*/
    /* Well, we need to allocate for entries for dlist */

    if (i >= MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc(sizeof(dltype));
      dlist[i]->defhdl = NULL;
    }

    /* Let's check the sizes and types for this select list item */

    switch (slist[i].dbtype) {

    case OCI_TYPECODE_NUMBER:

      /* The odescr will not give a good estimate to the scale if */
      /* no scale was given in the Oracle table definition.       */

#ifdef HAVE_SCALE
      if (slist[i].scale != 0) {
defbuf = (double *) dlist[i]->fbuf;
        deflen = FLT;
deftype = OCI_TYPECODE_DOUBLE;
slist[i].dbtype = OCI_TYPECODE_DOUBLE;
      } else {
defbuf = (int *) dlist[i]->ibuf;
        deflen = INT;
deftype = OCI_TYPECODE_INTEGER;
slist[i].dbtype = OCI_TYPECODE_INTEGER;
      }
#else
      defbuf = (double *) dlist[i]->fbuf;
      deflen = FLT;
      deftype = OCI_TYPECODE_FLOAT;
      slist[i].dbtype = OCI_TYPECODE_FLOAT;
#endif /* HAVE_SCALE */

      break;

    default:

      /* default is character string */

      defbuf = (char **) dlist[i]->sbuf;
      deflen = MAX_STR_LEN;
      deftype = SQLT_STR;
/*      deftype = OCI_TYPECODE_CHAR; */
      break;
    }

    /* Define the column */

    if ((status=OCIDefineByPos(curq,&(dlist[i]->defhdl),errhp,POS(i),
                                    defbuf,deflen,deftype,NULL,
                                    dlist[i]-
>rlen,NULL,OCI_DEFAULT))!=OCI_SUCCESS)
      sql_error(errhp,status,1);
  }
  return i;
}


/* process_select_list(): Fetch rows from a query.
*/

void process_select_list(num)
     int num;        /* number of select list items */
{

  int i,j;
  int ntf;
  int num_so_far;
  sword stats = OCI_SUCCESS;

  /* Print the headers for the query execution result */

  print_header(num);

  /* See if we need to limit the rows to fetch */

  ntf = (num_to_fetch >= 0) ? num_to_fetch : MAX_ARRAY;

  /* Fetch the rows and print them out */

  if ((ntf > MAX_ARRAY) || (num_to_fetch == -1)) {

    stats = OCIStmtFetch(curq, errhp, MAX_ARRAY, OCI_FETCH_NEXT,
OCI_DEFAULT);

    OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW_COUNT,err
hp);

    print_rows(num,rows_ret);

    /* To avoid 1022 from OFEN */
    /* More rows to fetch...   */

    if (stats != OCI_NO_DATA) {
      if (num_to_fetch == -1) {
while ((stats = OCIStmtFetch(curq,errhp,MAX_ARRAY,OCI_FETCH_NEXT,
                                          OCI_DEFAULT)) ==
OCI_SUCCESS) {
  OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                        OCI_ATTR_ROW_COUNT,errhp);
  print_rows(num,(num_so_far-rows_ret));
     rows_ret = num_so_far;
               }
  /* Print the final rows */
OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                        OCI_ATTR_ROW_COUNT,errhp);
print_rows(num,(num_so_far-rows_ret));
    rows_ret = num_so_far;
      } else {
```

```
      ntf -= MAX_ARRAY;

while ((stats = OCIStmtFetch(curq,errhp,
                                          ((ntf>MAX_ARRAY) ?
MAX_ARRAY:ntf),
                                          OCI_FETCH_NEXT,
OCI_DEFAULT)) ==
           OCI_SUCCESS) {
       ntf -= MAX_ARRAY;
  OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                    OCI_ATTR_ROW_COUNT,errhp);
  print_rows(num,(num_so_far-rows_ret));
     rows_ret = num_so_far;
      if (ntf <= 0) break;
             }
OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                     OCI_ATTR_ROW_COUNT,errhp);
print_rows(num,(num_so_far-rows_ret));
    rows_ret = num_so_far;
      }
    }
  } else {
    OCIStmtFetch(curq, errhp, ntf, OCI_FETCH_NEXT, OCI_DEFAULT);
    OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_ROW_COUNT,err
hp);
    print_rows(num,rows_ret);
  }

  fprintf(logfile,"\n\n%d %s processed.\n", rows_ret,
  rows_ret == 1 ? "row" : "rows");

}


int get_statement()
{

  char line[128];
  char *pos, *str;

  /* Reset statement buffer */

  stmt[0] = '\0';

  while (fgets(line, 127, qtemp) != NULL) {

    /* skip blank lines */
    if (line[0] == '\n')
      continue;

    /* remove blanks */

    str = line;

    while (*str == ' ') str++;

    /* Let's get the line together first */

    strcat(stmt, str);

    /* if this is a comment line */
    if ((str[0] == '-') && (str[1] == '-'))
      return COMMENT;

    /* see if this is a set_fetchrows line */
    if (strncmp(str, "set_fetchrows", 13) == 0) {
      pos = strchr(str, ';');
      *pos = '\0';
      pos = strchr(str, '=');
      num_to_fetch = atol(++pos);
      return SET_FETCHROW;
    }

    /* if this is the end of the current statement */
    if ((pos = strchr(stmt, ';')) != NULL) {
      *pos = '\0';
      return SQL_STMT;
    }
  }
  return END_OF_FILE;
}


/* memalloc(): Allocates memory, exit program if we have a problem.
*/

void *memalloc(size)
     int size;
{

  void *tmp;

  if ((tmp = (void *) malloc(size)) == NULL) {
    fprintf(stderr, "Error in malloc\n");
    SQLexit();
    return NULL;       /* should never reach here */
  } else {
    return tmp;
  }
}



void print_header(nsel)
     int nsel;           /* Number of select list items */
{

  int i, diff;
  char colname[MAX_COLNAME_SIZE];
  int len = 0;           /* Running column length */
  int cwid = 0;

  fprintf(logfile, "\n");

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

    /* extract the column name */

    strncpy((char *)colname, (char *)slist[i].buf, slist[i].buflen);
    colname[slist[i].buflen] = '\0';

    /* format the output a little */

    cwid = MAX(slist[i].dbsize, slist[i].buflen);

    /* do a little bit of formatting */

    if (cwid > 80) {
      fprintf(logfile,"\n");
      len = 0;
    } else if ((len += cwid) > 80) {
      fprintf(logfile,"\n");
      len = cwid;
    }
#ifdef FORMAT1
    if ((slist[i].dbtype == INT_TYPE) || (slist[i].dbtype ==
FLT_TYPE))
      fprintf(logfile, "%*s ", cwid, slist[i].buf);
    else  /* string type */
      fprintf(logfile, "%*s ", -cwid, slist[i].buf);
#else
      fprintf(logfile, "%*s ", -cwid, colname);
#endif /* FORMAT1 */
  }

  fprintf(logfile,"\n");
}


void print_rows(ncol, nrow)
     int ncol;
     int nrow;
{

  int i,j;
  int len;
  int diff;
  int cwid;

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

    len = 0;

    for (j=0;j<ncol;j++) {

      cwid = MAX(slist[j].dbsize, slist[j].buflen);

      /* do a little bit of formatting */

      if (cwid > 80) {
    fprintf(logfile,"\n");
          len = 0;
      } else if ((len += cwid) > 80) {
    fprintf(logfile,"\n");
         len = cwid;
      }

      switch(slist[j].dbtype) {
      case INT_TYPE:
#ifdef HAVE_SCALE
fprintf(logfile, "%*ld|", cwid,         (dlist[j]->ibuf)[i]);
            break;
#endif /* HAVE_SCALE */
      case FLT_TYPE:
#ifdef FORMAT1
fprintf(logfile,"%*.2f ", cwid, (dlist[j]->fbuf)[i]);
#else
fprintf(logfile,"%*.2f ", -cwid, (dlist[j]->fbuf)[i]);
#endif /* FORMAT1 */
            break;
      default:
fprintf(logfile, "%*s ", -(cwid), (dlist[j]->sbuf)[i]);
            break;
      }
    }
    fprintf(logfile, "\n");
```

```
  }
}


/* remove_newline(): Remove newline character from str. */

void remove_newline(str)
     char *str;
{

  char *p;

  while ((p = strchr(str,'\n')) != NULL)
    *p = ' ';
}


# ------------------------------------------------------------------
# qexecpl.h
# ------------------------------------------------------------------

/*
 * $Header: qexecpl.h 13-nov-2001.17:52:35 mpoess Exp $
 */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights
reserved.  */

/* NOTE:  See 'header_template.doc' in the 'doc' dve under the
'forms'
      directory for the header file template that includes
instructions.
*/

/*
   NAME
     qexecpl.h

   DESCRIPTION
     SQL statement execution front-end header file.

   PUBLIC FUNCTION(S)
     <list of external functions declared/defined - with one-line
descriptions>

   PRIVATE FUNCTION(S)
     <list of static functions defined in .c file - with one-line
descriptions>

   EXAMPLES

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   mpoess      11/13/01 - change DOP to 84 for DML and DDL
   mpoess      02/22/01 - add linux changes
   mpoess      08/05/99 - make compile
   mpoess      07/15/99 - Creation
   mpoess      07/15/99 - Creation

*/

/*
# ifndef S_ORACLE
#  include <s.h>
# endif
*/
#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

#include <oratypes.h>

#include <oratypes.h>

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

#ifndef OCI_ORACLE
#include <oci.h>
#endif /* OCI_ORACLE */
/*
#ifdef __STDC__
#include <ociapr.h>
#else
#include <ocikpr.h>
#endif *//* __STDC__ */

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

#ifndef TRUE
#define TRUE 1
#endif /* TRUE */

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */
#ifndef LINUX
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
#endif
/* defines and typedefs for parsing */

#define CRT_TBL  1
#define INS_STMT 3
#define SEL_STMT 4
#define UPD_STMT 5
#define DRP_VIEW 7
#define DRP_TBL  8
#define DEL_STMT 9
#define CRT_VIEW 10

/* defines and typedefs for query description */

#define MAX_COLNAME_SIZE 32   /* Maximum length of Column name  */
#define MAX_SEL_LIST 16       /* Maximum items on a select list */

#define END_OF_LIST 1007      /* Error code when we reach the end of
the */
                              /* select list.
*/

/* types for describe */

#define CHAR_TYPE 1
#define NUM_TYPE  2
#define INT_TYPE  3
#define FLT_TYPE  4
#define STR_TYPE  5
#define DATE_TYPE 12

#define NUMWIDTH 16           /* Width of the numeric fields    */

#define POS(i) (i+1)          /* The position is 1...n instead  */
#define IND(i) (i-1)          /* of 0..n-1 as in an array.      */

typedef struct des
{
  ub2 dbsize;
  ub4 buflen;
/*  sb2 dsize;  */
  sb4 scale;
/*  sb2 nullok; */
  OCITypeCode dbtype;
/*  text buf[MAX_COLNAME_SIZE]; */
  text *buf;
  ub1 precision;
} sltype;


/* defines and typedefs for query select list definition */

#define MAX_ARRAY 50       /* Maximum array size for array fetch */
#define PFMEMSIZE 65536    /* Memory size of prefetch buffer     */

#define MAX_STR_LEN 256    /* Maximum size for string variables
*/
#define MAX_PREALLOC 8     /* Maximum number of preallocated select
list */
                           /* definitions.
*/

#define INT sizeof(long)
#define STR sizeof(char)
#define FLT sizeof(double)

#define FLTP (double *)
#define INTP (long *)
#define STRP (char **)

typedef struct def
{
  long ibuf[MAX_ARRAY];
  double fbuf[MAX_ARRAY];
  char sbuf[MAX_ARRAY][MAX_STR_LEN];
  ub2 rlen[MAX_ARRAY];         /* return length     */
  OCIDefine *defhdl;
} dltype;

extern int errno;

#define SQL_LEN 2048
```

```
#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

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

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define SID(sid) ((sid == -1) ? 0 : sid)

/* For get_statement */

#define END_OF_FILE -1
#define COMMENT 1
#define SQL_STMT   2
#define SET_FETCHROW 3


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

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

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

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

#define OCIsexec(svch,stmh,errh,iter) \
    if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI_DEF
AULT)) != OCI_SUCCESS) \
       sql_error(errh,status,1); \
    else \
       DISCARD 0


#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree
84)"
#define PDDLTXT "alter session force parallel ddl parallel (degree
84)"

#endif /* QSTREAMPL_H */


# ---------------------------------------------------------------
# ri.sql
# ---------------------------------------------------------------

select count(*) from customer left outer join nation on (c_nationkey
= n_nationkey) where n_nationkey is null;
select count(*) from lineitem left outer join orders on
(l_orderkey=o_orderkey) where o_orderkey is null;
select count(*) from lineitem left outer join partsupp on
(l_partkey=ps_partkey and l_suppkey=ps_suppkey) where (ps_partkey is
null or ps_suppkey is null);
select count(*) from nation left outer join region on
(n_regionkey=r_regionkey) where r_regionkey is null;
select count(*) from orders left outer join customer on
(o_custkey=c_custkey) where c_custkey is null;
select count(*) from partsupp left outer join part on (ps_partkey =
p_partkey) where p_partkey is null;
select count(*) from partsupp left outer join supplier on (ps_suppkey
= s_suppkey) where s_suppkey is null;
select count(*) from supplier left outer join nation on (s_nationkey
= n_nationkey) where n_nationkey is null;
exit;


# ---------------------------------------------------------------
# runTPCH
# ---------------------------------------------------------------

#!/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`
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=${OUT_DIR}/firstten
RI_OUT=${OUT_DIR}/ri_out
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop


#2starts  >> $SCRIPT_LOG_FILE
ckpnt.sh
echo "Start: dbtables.sql, count.sql and ri.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
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/ri > ${RI_OUT} 2>&1
echo "End: dbtables.sql, count.sql and ri.sql" >> $SCRIPT_LOG_FILE
runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}
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


# ---------------------------------------------------------------
# runTPCH_build
# ---------------------------------------------------------------

#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

sqlplus=$ORACLE_HOME/bin/sqlplus
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

if [ ! -f $RUN_ID_FILE ]
then
  echo "0" > $RUN_ID_FILE
fi

RUN_ID=`cat $RUN_ID_FILE`
RUN_ID=`expr $RUN_ID + 1`
echo $RUN_ID > $RUN_ID_FILE

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop
```

```
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
#dbcre_10gR2.sh >> $LD1DBCRE
#tscre_10gR2.sh >>  $LD2SCTSO

#stopdb
#wait
#startdb
#wait
STIME=`$GTIME`
SLTIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
/home/oracle/kit/audit/dapop_final.sh >> $LD3DAPOP
$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE
echo Generated seed: `cat $KIT_DIR/audit/seed` >> $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=/home/oracle/dbgen241
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY
export DSS_CONFIG=/home/oracle/dbgen241

UPD_SQL=${UPD_DIR}/sql
UPD_SPT=${UPD_DIR}
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


# ---------------------------------------------------------------
# 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}
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
# ---------------------------------------------------------------

#!/bin/ksh
#
# $Header: runuf1.sh 25-oct-2001.15:56:04 mpoess Exp $
#
# runuf1.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. All rights reserved.
#
# NAME
# runuf1.sh - <one-line expansion of the name>
```

```
#
# DESCRIPTION
# runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
# -p [<program>] <run_id> <scale factor> <pair number>
# <parallelism>
# USAGE
# To execute UF1.
#
# NOTES
# <other useful comments, qualifications, etc.>
#
# MODIFIED (MM/DD/YY)
# mpoess 10/25/01 - change default directory for update sets
# mpoess 10/17/01 - add support for external tables
# mpoess 08/15/99 - Creation
# mpoess 08/15/99 - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
#PAR_HINT=${UPDATE_DOP}
PAR_HINT=128
LOGPATH=.
PASSWD=tpch/tpch
if [ $# -lt 1 ];
then
echo runuf1.sh setnum
drop table temp_o_et;
drop table temp_l_et;
exit 1
fi
SETNUM=$1
i=1
PID=""
# perform the update function 1
START=`$GTIME`
# first create the temp tables
sqlplus /NOLOG << !
connect tpch/tpch;
set timing on
set serveroutput on
set echo on
drop directory data_dir;
create directory data_dir as '/data01/update';
drop directory du1;
create directory du1 as '/data01/update';
drop directory du2;
create directory du2 as '/data01/update';
drop directory du3;
create directory du3 as '/data01/update';
drop directory du4;
create directory du4 as '/data01/update';
drop directory du5;
create directory du5 as '/data01/update';
drop directory du6;
create directory du6 as '/data01/update';
drop directory du7;
create directory du7 as '/data01/update';
drop directory du8;
create directory du8 as '/data01/update';

create table temp_l_et(
l_orderkey number ,
l_partkey number ,
l_suppkey number ,
l_linenumber number ,
l_quantity number ,
l_extendedprice number ,
l_discount number ,
l_tax number ,
l_returnflag char(1) ,
l_linestatus char(1) ,
l_shipdate date ,
l_commitdate date ,
l_receiptdate date ,
l_shipinstruct char(25) ,
l_shipmode char(10) ,
l_comment varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
records delimited by newline
nobadfile
nologfile
fields terminated by '|'
missing field values are null
)
location (
du1:'lineitem.tbl.u${SETNUM}.1',
du2:'lineitem.tbl.u${SETNUM}.2',
du3:'lineitem.tbl.u${SETNUM}.3',
du4:'lineitem.tbl.u${SETNUM}.4',
du5:'lineitem.tbl.u${SETNUM}.5',
du6:'lineitem.tbl.u${SETNUM}.6',
du7:'lineitem.tbl.u${SETNUM}.7',
du8:'lineitem.tbl.u${SETNUM}.8'
))
reject limit unlimited;
drop table temp_o_et;

create table temp_o_et(
o_orderkey number ,
o_custkey number ,
o_orderstatus char(1) ,
o_totalprice number ,
o_orderdate date ,
o_orderpriority char(15) ,
o_clerk char(15) ,
o_shippriority number ,
o_comment varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
records delimited by newline
nobadfile
nologfile
fields terminated by '|'
missing field values are null
)
location (
du1:'orders.tbl.u${SETNUM}.1',
du2:'orders.tbl.u${SETNUM}.2',
du3:'orders.tbl.u${SETNUM}.3',
du4:'orders.tbl.u${SETNUM}.4',
du5:'orders.tbl.u${SETNUM}.5',
du6:'orders.tbl.u${SETNUM}.6',
du7:'orders.tbl.u${SETNUM}.7',
du8:'orders.tbl.u${SETNUM}.8'
))
reject limit unlimited;
alter table temp_l_et parallel 64;
alter table temp_o_et parallel 64;
alter session force parallel dml parallel (degree ${PAR_HINT});
alter session set isolation_level = serializable;
alter session set optimizer_index_cost_adj = 1;
insert into orders (
select
o_orderdate ,
o_orderkey ,
o_custkey ,
o_orderpriority ,
o_shippriority ,
o_clerk ,
o_orderstatus ,
o_totalprice ,
o_comment
from temp_o_et);
insert into lineitem (
select
l_shipdate ,
l_orderkey ,
l_discount ,
l_extendedprice ,
l_suppkey ,
l_quantity ,
l_returnflag ,
l_partkey ,
l_linestatus ,
l_tax ,
l_commitdate ,
l_receiptdate ,
l_shipmode ,
l_linenumber ,
l_shipinstruct ,
l_comment
from temp_l_et);
commit;
drop table temp_l_et;
drop table temp_o_et;
exit;
!
END=`$GTIME`
# Done
echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""


# ------------------------------------------------------------------
# runuf2.sh
# ------------------------------------------------------------------

#!/bin/ksh
```

```
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation. All rights reserved.
#
# NAME
# runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
# runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
# <scale factor> <pair number> <parallelism>
# USAGE
# To execute UF2.
#
# NOTES
# <other useful comments, qualifications, etc.>
#
#
. $KIT_DIR/env
UPDATE_DIR=${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_HINT_ET=32
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 tpch/tpch;
set timing on
set serveroutput on
set echo on
drop directory data_dir;
create directory data_dir as '/data01/update';
drop directory du1;
create directory du1 as '/data01/update';
drop directory du2;
create directory du2 as '/data01/update';
drop directory du3;
create directory du3 as '/data01/update';
drop directory du4;
create directory du4 as '/data01/update';
drop directory du5;
create directory du5 as '/data01/update';
drop directory du6;
create directory du6 as '/data01/update';
drop directory du7;
create directory du7 as '/data01/update';
drop directory du8;
create directory du8 as '/data01/update';

create table temp_okey_et(
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 (
du1:'delete.u${SETNUM}.1',
du2:'delete.u${SETNUM}.2',
du3:'delete.u${SETNUM}.3',
du4:'delete.u${SETNUM}.4',
du5:'delete.u${SETNUM}.5',
du6:'delete.u${SETNUM}.6',
du7:'delete.u${SETNUM}.7',
du8:'delete.u${SETNUM}.8'))
reject limit unlimited;
alter table temp_okey_et parallel ${PAR_HINT_ET};
create table temp_okey_${SETNUM} nologging as
select * from temp_okey_et;
alter table temp_okey_${SETNUM} parallel ${PAR_HINT};
create unique index i_temp_okey_${SETNUM} on temp_okey_${SETNUM}
(t_orderkey)
parallel ${PAR_HINT} nologging compute statistics;
analyze table temp_okey_${SETNUM} estimate statistics sample 2
percent;
alter session force parallel dml parallel 128;
alter session set isolation_level=serializable;
alter session set optimizer_index_cost_adj=1;
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;
drop table temp_okey;
drop table temp_okey_et;
exit;
!
END=`$GTIME`
# Done
echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

Appendix G: 3rd Party Price Quotations

From: Mary.Beth Pierantoni [mailto:mary.beth.pierantoni@oracle.com]
Sent: Wednesday, September 06, 2006 4:59 PM
To: bob@pantasys.com
Subject: Pricing

To following is pricing for 32 processors/64 cores, Oracle Database Enterprise Edition with Oracle Real Application Clusters and Partitioning.

Product	Price	Quantity	Extended Price
Oracle Database 10g Enterprise Edition Release 2, Named User Plus for 3 years	10,000	32*	320,000
Oracle Real Application Clusters Named User Plus for 3 years	5,000	32*	160,000
Partitioning Named User Plus for 3 years	2,500	32*	80,000
Database Server Support Package	16,000	3	48,000
Oracle Mandatory E-Business Discount			<121,600>
TOTAL			486,400

Oracle Pricing Contact: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 916-315-5081
** 32 = 64 * 0.50. Explanation: For the purposes of counting the number of processors which require licensing, an AMD multicore chip with “n” cores shall be determined by multiplying “n” cores by a factor of 0.50.