TPC Benchmark™ H
Full Disclosure Report

Sun Microsystems Starfire™ Enterprise™ 10000 Server Using Oracle 9i Database Enterprise Edition 9.0.1

Submitted for Review
Report Date: April 13, 2001

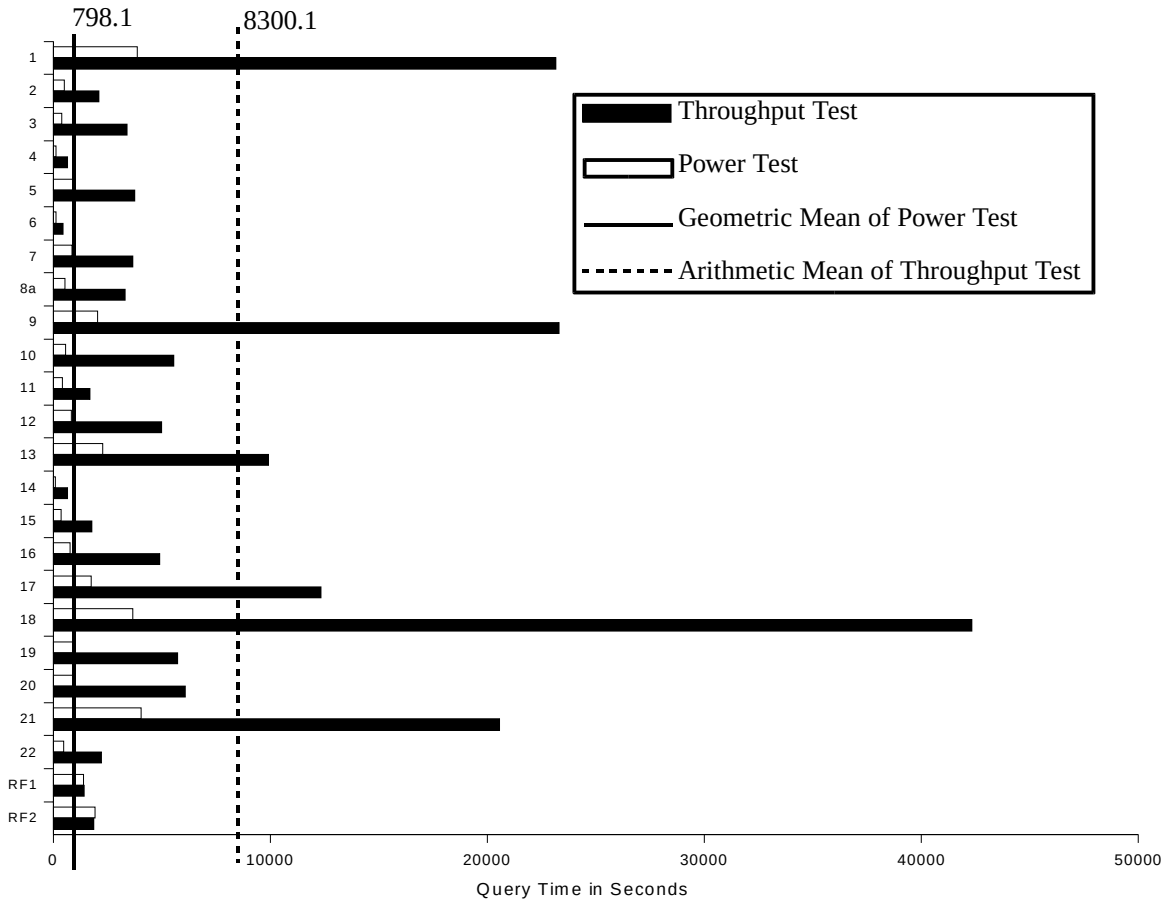
First Printing

901 San Antonio Road, Palo Alto, California 94303 U.S.A.

Sun microsystems / ORACLE	**Sun Starfire Enterprise 10000 with Oracle9i**	**TPC−H Rev. 1.3.0**
		April 13, 2001

Total System Cost	Composite Query per Hour Metric	Price/Performance
$13,155,003	**10,764.7** QphH@3000GB	**$1222** $/QphH@3000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
3000GB	Oracle9i Database Enterprise Edition 9.0.1	Solaris 8	Veritas Volume Mgr. 3.0.4e	June 19, 2001*



Database Load Time = 14:58	Load Includes Backup: N	Total Data Storage/Database Size = 8.14
RAID (Base tables): N	RAID (Base tables and auxiliary data structures): Y	RAID (All): N

System Configuration: 2 node Starfire Enterprise 10000 cluster
Processors: 2x64 400 Mhz UltraSPARC II CPUs each with 8MB E−Cache
Memory: 2x64GB memory
Disks: 60 A5200 (22x18.2GB disks), 8 T3 (9x18.2GB disks), 4 D1000 (12x18.2GB)
Total Storage: 24408.1GB (In this calculation one GB is defined as 1024*1024*1024 bytes)

* All Hardware and System Software available immediately. Database software available June 19, 2001

Sun Starfire Enterprise 10000 with Oracle9i

TPC−H Rev. 1.3.0

April 13, 2001

Pricing Summary

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5 Yr. Maint.
Server Hardware						
E10000 Base cabinet	E10000−4	2	133533	2	267065	675840
System Board unpopulated	2761A	2	40802	32	1305652	1162752
CPU 400 MHz 8MB L2$	2580A	2	11087	128	1419131	
Memory Board unpopulated	7025A	2	7391	32	236512	
1 GB for Ex000	7023A	2	6061	128	775791	
Dual SBus I/O daughter card	2730A	2	5543	26	144130	
PCI I/O daughter card	2731A	2	7418	6	44511	
System Service Processor	2755A	2	11457	2	22913	6528
AC input module	3875A	2	2217	8	17739	
48 volt power supply	9685A	2	2217	16	35478	
Power Control Module	9681A	2	742	2	1484	
Fan Tray	9671A	2	1109	32	35478	
Control Board w/Enet Hub	2722A	2	11087	2	22174	
Power Cord for System	3865A	2	0	8	0	
FCAL Host Adapter	6730A	2	1996	96	191582	
SCI Host Adapter	X1074A	2	3326	12	39913	
PCI−SCI Cable 5m	X3902A	2	0	12	0	
Sbus QFE 2.0 Host Adapter	1049A	2	1475	4	5898	
SBus Utlra DWIS/S Host Adapter	1065A	2	957	4	3829	
Power Cord for Data Center Cab (US)	3800A	2	0	2	0	
12 Meter SCSI cable	979A	2	288	4	1152	
15M Fibre Channel Cable	978A	2	190	56	10652	
North American Country Kit	3508A		0	2	0	
Subtotal					4581095	1845120
SunService Discount (10% Volume + 5% yearly prepayment)						−267542
Server Hardware Subtotal					4581095	1557578
Storage						
Rack Mount D1000 W/12x18GB	SG−ARY154A−218GR5	2	12554	4	50217	
1310 GB Sun StorEdge T3 ES	XT3ES−RK−88−1310	2	167857	1	167857	99264
2400GB Sun StorEdge A5200	SG−ARY543A−2400G	2	229500	10	2295000	1184400
U.S. Power Cord for StorEdge	X3858A	2	0	20	0	
Subtotal					2513074	1283664
SunService Discount (10% Volume + 5% yearly prepayment)						−186131
Storage Subtotal					2513074	1097533
Server Software						
Solaris 8 Std Media	SOLZS−080B9AYS	1	54	2	108	
SPARC Compiler C/C++	FCEIS−601−T999	1	3495	2	6990	2760
Sun StorEdge Comp Mgr	SCMMS−210−9P99	2	0	2	0	
Veritas VM 3.0.4 for A5x00s	VVMGS−304−9999	2	0	2	0	
Sun Cluster 2.2 for E10000	CL0VS−22A−9999	1	50000	2	100000	184800
E10000 SSP SW 3.3	SSP9S−330−SAM9	2	0	2	0	
Oracle 9i Database Enterprise Edition 9.0.1 w/ Real Application Clusters and Partitioning Options		3	1330560	1	1330560	2090880
Subtotal					1437658	2278440
SunService Discount (10% Volume + 5% yearly prepayment)						−330374
Server Software Subtotal					1437658	1948066
			Total		8531827	4623177
Service for all Sun products is from Sun Microsystems, Inc.			5 Yr. Cost		13155003	
Service for Oracle products is from Oracle Corp.			QphH@3000GB		10764.7	
			$/QphH@3000GB		1222.05	

Notes (Source):
1. Sun Microsystems, Inc.
2. CAT Technology, Inc.
3. Oracle Corp.

Audited by: François Raab, InfoSizing, Inc. (www.sizing.com)

Sun microsystems ORACLE	**Sun Starfire Enterprise 10000 with Oracle9i**	**TPC−H Rev. 1.3.0**
		April 13, 2001

Numerical Quantities

Measurement Results:

Database Scale Factor	= 3000
Total Data Storage / Database Size	= 8.14
Start of database load time	= 2001−03−23 15:40:39
End of database load time	= 2001−03−24 06:37:52
Database Load Time	= 14:58
Query Streams for Throughput Test	= 8
TPC−H Power	= 13,532.3
TPC−H Throughput	= 8,563.2
TPC−H Composite Query−per−Hour Rating (QphH@3000GB)	= 10,764.7
Total System Price Over 5 Years	= $13,155,003
TPC−H Price/Performance Metric ($/QphH@3000GB)	= $1222

Measurement Intervals:

Measurement Interval in Throughput Test (Ts)	= 221,973 seconds

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Duration
Stream 00	324063752	3/24/01	10:28:06	3/24/01	18:46:40	8:18:34
Stream 01	324063753	3/24/01	18:46:41	3/26/01	17:43:26	46:56:45
Stream 02	324063754	3/24/01	18:46:41	3/27/01	01:07:30	54:20:49
Stream 03	324063755	3/24/01	18:46:41	3/26/01	15:38:48	44:52:07
Stream 04	324063756	3/24/01	18:46:41	3/26/01	23:26:39	52:39:58
Stream 05	324063757	3/24/01	18:46:41	3/26/01	23:15:43	52:29:02
Stream 06	324063758	3/24/01	18:46:41	3/26/01	20:29:09	49:42:28
Stream 07	324063759	3/24/01	18:46:41	3/26/01	22:34:04	51:47:23
Stream 08	324063760	3/24/01	18:46:41	3/26/01	23:27:21	52:40:40
Refresh		3/27/01	01:07:32	3/27/01	08:26:14	7:18:42

Sun Starfire Enterprise 10000 with Oracle9i

TPC−H Rev. 1.3.0

April 13, 2001

TPC−H Timing Intervals (in seconds)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 00	3879.2	523.7	382.3	134.3	934.4	125.6	851.8	528.0	2057.0	572.3	413.4	839.0
Stream 01	20199.8	2042.5	4211.1	502.7	2355.7	378.7	3160.7	3482.1	21094.4	3444.6	1479.4	3080.3
Stream 02	23876.6	2092.5	683.0	376.2	4527.5	198.4	4444.2	3396.8	23141.4	3918.5	948.1	5132.9
Stream 03	21290.2	2142.2	4201.8	688.1	3750.2	539.7	3461.8	3010.2	14271.3	2750.9	1737.9	5563.8
Stream 04	24365.8	2191.8	2806.9	754.3	2780.9	625.0	2268.6	3087.8	23310.0	3785.2	1764.7	5198.5
Stream 05	23826.7	2303.6	4612.9	881.1	4208.6	522.9	4673.4	3345.4	14902.5	2106.6	2349.4	5242.6
Stream 06	26494.2	2409.1	2540.8	791.4	4342.1	555.4	4455.1	2876.0	21645.3	1450.6	1986.7	4788.9
Stream 07	24324.8	2111.3	3613.3	515.3	3559.0	360.1	2658.0	3896.0	45423.3	2487.6	1978.4	4267.3
Stream 08	21000.1	1544.0	4478.8	805.9	4582.2	473.1	4109.2	3329.8	22552.1	24578.6	1193.2	6799.4
Minimum	20199.8	1544.0	683.0	376.2	2355.7	198.4	2268.6	2876.0	14271.3	1450.6	948.1	3080.3
Average	23172.3	2104.6	3393.6	664.4	3763.3	456.7	3653.9	3303.0	23292.6	5565.3	1679.7	5009.2
Maximum	26494.2	2409.1	4612.9	881.1	4582.2	625.0	4673.4	3896.0	45423.3	24578.6	2349.4	6799.4

	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	2294.8	107.9	353.6	772.6	1737.6	3669.9	917.9	970.5	4056.4	474.2	1398.9	1918.9
Stream 01	11176.5	801.5	1946.4	4692.1	6795.8	35182.2	5850.0	5045.5	30791.2	2359.2	1477.2	1878.1
Stream 02	12804.7	683.2	1469.0	5048.9	22742.5	62743.4	5720.8	2925.6	6354.3	2419.4	1421.8	1853.2
Stream 03	10358.6	478.2	2079.9	5077.2	10334.9	31010.5	5552.1	9142.2	22274.7	1809.8	1372.2	1854.7
Stream 04	7012.7	698.4	1294.5	5566.1	12955.0	59530.5	5137.1	2917.2	20173.4	1373.3	1411.0	1882.9
Stream 05	12617.7	820.1	2341.6	5045.7	9295.7	39479.5	5742.9	7841.4	33988.0	2793.2	1403.5	1881.8
Stream 06	8931.2	770.9	1763.9	5732.8	12025.4	42579.7	5565.2	7269.8	17473.6	2498.9	1390.5	1845.1
Stream 07	10203.0	292.2	1018.8	2925.2	11855.9	31486.6	5280.8	6393.0	19816.0	1976.5	1439.1	1868.9
Stream 08	6132.3	747.4	2293.3	5253.8	12705.2	36554.0	7044.8	7249.4	13745.8	2466.6	1456.2	1881.6
Minimum	6132.3	292.2	1018.8	2925.2	6795.8	31010.5	5137.1	2917.2	6354.3	1373.3	1372.2	1845.1
Average	9904.6	661.5	1775.9	4917.7	12338.8	42320.8	5736.7	6098.0	20577.1	2212.1	1421.4	1868.3
Maximum	12804.7	820.1	2341.6	5732.8	22742.5	62743.4	7044.8	9142.2	33988.0	2793.2	1477.2	1882.9

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsors:

Brad Carlile
Manager, Strategic Apps. Eng.
Sun Microsystems, Inc.
8300 SW Creekside Place
Beaverton, OR 97008

Ray Glasstone
Manager DSS Performance
Oracle Corporation
100 Oracle Parkway
Redwood Shores, CA 94065

April 13, 2001

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

Platform: **Sun Starfire Enterprise 10000**
Database Manager: **Oracle9i**
Operating System: ***Solaris 8***

The results were:

CPU (Speed)	Memory	Disks	**QphH@3000GB**
2 Nodes Sun Starfire Enterprise 10000 (each with)			
64 x UltraSPARC II (400 MHz)	8MB E–Cache/cpu 64 GB Main	60 x 22 x 18.2 GB 8 x 9 x 18.2 GB 4 x 12 18.2 GB	**10764.7**

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
- The database was properly scaled to 3000GB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met

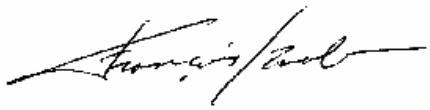
1373 North Franklin Street – Colorado Springs, CO 80903–2527 – Office: 719/473–7555 – Fax: 719/473–7554

- The query input variables were generated by QGEN
- The query text was produced using minor modifications and an approved 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
- 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

1373 North Franklin Street – Colorado Springs, CO 80903–2527 – Office: 719/473–7555 – Fax: 719/473–7554

Table of Contents

TPC Benchmark H Overview

The TPC BenchmarkTM H (TPC−H) is a Decision Support benchmark. It is a suite of business−oriented ad−hoc queries and concurrent modifications. The queries and the data populating the database have been chosen to have broad industry−wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates Decision Support systems that:

- Examine large volumes of data
- Execute queries with a high degree of complexity
- Give answers to critical business questions

TPC−H evaluates the performance of various Decision Support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC−H queries:

- Give answers to real−world business questions
- Simulate generated ad−hoc queries
- Are far more complex than most OLTP transactions
- Include a rich breadth of operators and selectivity constraints
- Generate intensive activity on the part of the database server component of the system under test
- Are executed against a database complying to specific population and scaling requirements
- Are implemented with constraints derived from staying closely synchronized with an on−line production database

1. General Items

1.1 Benchmark Sponsor

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

Sun Microsystems, Inc. and Oracle Corp. are the sponsors of this TPC−H benchmark.

1.2 Parameter Settings

Settings must be provided for all customer−tunable parameters and options that 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*

Appendix A contains the Solaris and Oracle parameters used in this benchmark.

1.3 Configuration Diagram

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences.

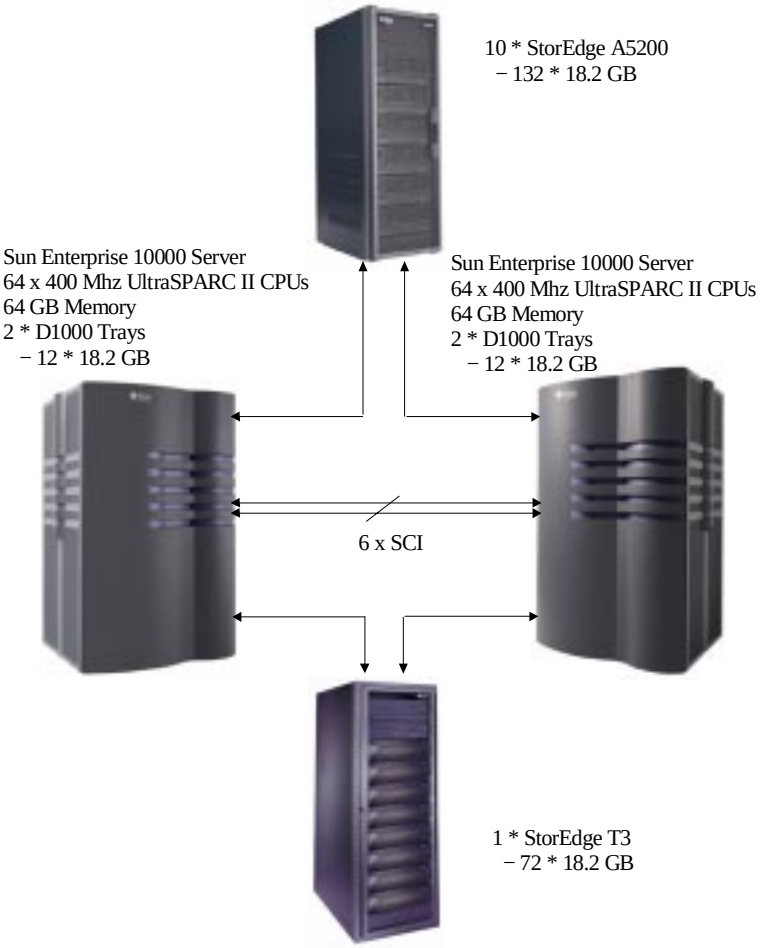
Two Sun Enterprise 10000 Servers, each configured with:

- 64 400 MHz UltraSPARC II CPUs each with 8MB E−Cache
- 64 GB memory
- 2 D1000 disk arrays each containing 12 18.2GB disk drives

- 1 Ethernet controller

The 2 servers are clustered via 6 dedicated SCI networks. The servers share the following disks:

- 60 A5200 disk arrays, each containing 22 18.2GB disk drives
- 8 T3 disk arrays, each containing 9 18.2GB disk drives



Tested and priced configurations are identical.

2. Clause 1 Logical Database Design

2.1 Database Definition Statements

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

Appendix B contains the programs and scripts that create and analyze the tables and indexes 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.

No record clustering or index clustering was used. Default column ordering was used. Refer to the table create statements in Appendix B for further details.

2.3 Horizontal Partitioning

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

Horizontal partitioning was used for all tables except NATION, REGION and SUPPLIER. Horizontal partitioning was used for all indexes except I_O_ORDERKEY. Refer to the table/index create statements in Appendix B for more details.

2.4 Replication

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

No replication was used.

3. 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 language used to implement all queries.

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 1.3.0 of DBGEN and QGEN were used for this TPC−H benchmark. DBGEN output routines were modified to separate the load data into multiple files. No changes were made to the actual data generation or random functions. For DBGEN changes see Appendix B.

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.

The supplied QGEN version 1.3.0 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 definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.

Appendix C contains the qualification 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 D 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 isolation 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 driver code included in Appendix E.

4. 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.

Source code for the ACID test is included in Appendix B.

4.2 Atomicity

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

4.2.1 Completed Transaction

Perform the 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

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 Aborted Transaction

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.

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

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

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.

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

4.4 Isolation

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

4.4.1 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.

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 the 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 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.

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 Write−Write Conflict with Commit

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

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

4.4.4 Write−Write Conflict with Rollback

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

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

4.4.5 Concurrent Progress of Read and Write Transactions

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

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to ROLLBACK.
2. Another Transaction, T2, was started which did the following:

   For random values of PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal, are returned.
3. T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that appropriate rows in ORDERS, LINEITEM and HISTORY tables were changed.

4.4.6 Read−Only Query Conflict with Update Transaction

Demonstrate that 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, executing 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

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

4.5.1 Failure of a 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 tables and log files were mirrored. During the durability test the disk containing one side of a data file mirror was removed from its cabinet. Similarly the disk containing one side of a log file mirror was removed from its cabinet. The test continued uninterrupted, using the remaining side of the mirror.

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 both servers was turned off simultaneously by flipping breakers at the main electrical panel 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 successfully.

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 section 4.5.2.

5. Clause 4 Scaling and Database Population

5.1 Ending Cardinality of Tables

The cardinality (i.e., 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	Rows
Orders	4,500,000,000
Lineitem	18,000,048,306
Customer	450,000,000
Part	600,000,000
Supplier	30,000,000
Partsupp	2,400,000,000
Nation	25
Region	5

5.2 Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described.

- Except for NATION and REGION, all tables and indexes were mirrored and striped logically across 1320 disks in the A5200 arrays. NATION and REGION were placed in the user default tablespace. The SYSTEM tablespace was striped and mirrored across 220 disks in the A5200 arrays.
- The 72 disks of the T3 arrays were used for the redo logs. Each group of 9 disks was striped into a RAID0 stripe and the logs were striped across 4 of these RAID0 stripes and then mirrored to a stripe across the other 4 RAID0 stripes.
- The temp tablespace was striped across 1320 disks in the A5200 arrays but not mirrored.

For more details refer to disk configuration section in Appendix B.

5.3 Database partition/replication mapping

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

The database was not replicated.

Horizontal partitioning was used for base tables LINEITEM, ORDERS, PARTSUPP, PART and CUSTOMER and for index I_O_ORDERKEY. The details for this partitioning can be understood by examining the syntax of the table and index definition statments in Appendix B.

5.4 RAID Feature

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.

Table/Index	RAID type
tables	RAID 0+1
indexes	RAID 0+1
temp tablespace	RAID 0
log	RAID 0+1
System tablespace	RAID 0+1

5.5 Modifications to the DBGEN

Any modifications to the DBGEN (see Clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The supplied DBGEN version 1.3.0 was used to generate the database population for this benchmark. DBGEN output routines were modified to separate the load data into multiple files. No changes were made to the actual data generation or random functions. For DBGEN changes see Appendix B.

5.6 Database Load Time

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

The database load time was 14 hours 58 minutes.

5.7 Data Storage Ratio

The data storage ratio must be disclosed. It is computed as the ratio between the total amount of priced disk space, and the chosen test database size as defined in Clause 4.1.3.

The data storage ratio is computed from the following information:

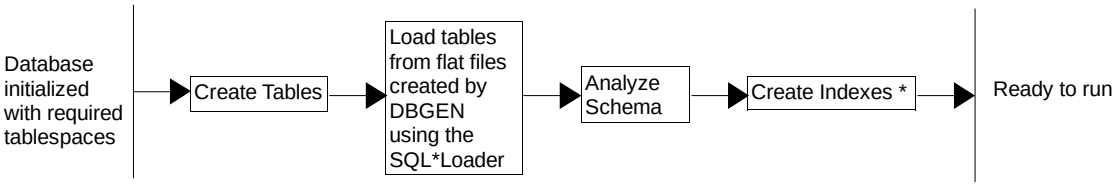
** Disk manufacturer definition of one GB is 10^9 bytes*
***In this calculation one GB is defined as 2^30 bytes*

Disk Type	# Of Disks	Space Per Disk*	Sub−Total Disk Space**
A5200	1320	18.2 GB	22374.1 GB
T300	72	18.2 GB	1220.4 GB
D1000	48	18.2 GB	813.6 GB
		Total Space	**24408.1 GB**
		Data Storage Ratio	**8.14**

5.8 Database Load Mechanism Details and Illustration

The details of the database load must be described, including a block diagram illustrating the overall process.

The database was loaded using data generation stored on flat files all on the tested and priced configurations. The SQL*Loader read the files that were created by the DBGEN program



* Analyze index performed during index creation

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 adjustments for the size difference.

6. Clause 5 Performance Metrics and Execution Rules

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.

1. Auditor requested queries were run against the database to verify the correctness of the database load
2. The database was brought down on both nodes
3. Both nodes were rebooted
4. The Sun Cluster software was brought up on both nodes
5. The database was restarted on both nodes

All scripts and queries used are included in Appendix F

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 for each query and for both refresh functions must be reported for the power test.

The power test timing intervals are:

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8a	Q9	Q10	Q11	Q12
Stream 00	3879.2	523.7	382.3	134.3	934.4	125.6	851.8	528.0	2057.0	572.3	413.4	839.0

	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	2294.8	107.9	353.6	772.6	1737.6	3669.9	917.9	970.5	4056.4	474.2	1398.9	1918.9

6.4 Number of Streams for the Throughput Test

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

Eight streams were used for the throughput test.

6.5 Start and End Date/Times for 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 contained in the Numerical Quantity Summary earlier in this document.

6.6 Total Elapsed Time of the Measurement Interval

The total elapsed time of the measurement interval must be reported for the throughput test.

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

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

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

The start and finish times for each refresh function in the refresh stream are contained in the Numerical Quantity Summary earlier in this document.

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

The timing intervals for each query of each stream and each refresh function must be reported for the throughput test.

The timing intervals for each query and each refresh function for the throughput test are contained in the Numerical Quantity 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, are contained in the Numerical Quantity Summary earlier in this document.

6.10 The Performance Metric and Numerical Quantities from Both Runs

The performance metric and numerical quantities from both runs must be disclosed.

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

Run ID	QppH@100GB	QthH@100GB	QphH@100GB
Run 1	13,532.3	8,563.2	10,764.7
Run 2	13,494.7	9,197.6	11,140.9
Difference	0.28%	6.89%	3.38%

6.11 System Activity Between Performance Tests

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

Both systems were rebooted and the database was restarted on both nodes between the two runs.

7. Clause 6 SUT and Driver Implementation

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.

The Power Test and Throughput Test are performed by a shell script called audit.sh. This calls audit_stream.sh to do a full power test and then throughput test. QGEN is first called with a stream id of 0 to generate the QET for the Power Test. UF1 is then started by executing the runuf1.sh script. Query submission follows, with the qexecpl.c ISL program. The execution of the UF2 script runuf2.sh rounds out the Power Test execution. Both wall−clock and high−resolution times are collected for all measurement intervals.

Following the Power Test, QGEN is again called with the subsequent 8 stream ids to generate new QET for each Throughput Test. qexecpl.c is called simultaneously for all 8 streams to execute the queries as above.Then the update_stream.sh script is called to run all 8 update pairs to finish the throughput run.

7.2 Implementation−Specific Layer

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.

Query execution text generated by QGEN is picked up by the ISL program which submits the query to the SUT.

The ISL program (qexecpl.c) utilizes the Oracle Call Interface (OCI) to communicate with the Oracle database on the SUT. EQTs directly generated by QGEN are read and submitted to the SUT via the ISL program (qexecpl.c) as dynamic SQL statements. The ISL program then fetches the query execution output and reports it to the user. Timings are taken at intervals specified in Section 5.3.7 of the TPC−H benchmark specification.

For the Update Functions, multiple sqlldr processes loaded data into temporary tables. Oracle9i's parallel insert and delete functionality was used to perform the Update Functions, selecting data from the temporary tables.

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 was not used.

8. Clause 7 Pricing

8.1 Hardware and Software Used

A detailed list of hardware and software used in the priced system must be reported. Each item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery date. If package−pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.

Refer to the Executive Summary.

8.2 Total Five Year Price

The total 5−year price of the entire configuration must be reported, including hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The total 5−year price of the configuration is $13,155,003. For details of pricing, see the second page of the Executive Summary.

Hardware prices are from reseller price quotes. Refer to Appendix G.

8.3 Availability Date

The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

Hardware and System software components are available now. Oracle software will be available by June 19, 2001.

9. Auditor's Information and Attestation Letter

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.

The auditor's attestation letter is included at the front of this report.

Appendix A. Solaris and Oracle9i Parameters

This Appendix contains Solaris kernel parameters and environment variables and Oracle initialization parameters.

==========

Oracle9i Parameters

(altered from default)

==========

tkinit.ora

==========

```
always_anti_join                 = HASH
always_semi_join               = HASH
audit_trail                      = FALSE
compatible                       = 8.1.7
control_files                    =
/dev/vx/rdsk/photon3/cntrl_1
db_block_buffers                 = 122020
db_block_lru_latches           = 64
db_block_max_dirty_target     =  0
db_block_size                    = 8192
db_file_multiblock_read_count   = 64
db_files                         = 1023
db_name                          = inst1
db_writer_processes              = 10
dml_locks                        = 80000
enqueue_resources                = 50000
distributed_transactions         = 0
fast_start_io_target           = 0
gc_files_to_locks                ="2-350=10EACH"
gc_rollback_locks                ="0-400=32!8REACH"
global_names                     = FALSE
hash_area_size                  = 70000000
hash_multiblock_io_count         = 32
java_pool_size                 = 0
max_dump_file_size               = 100000
max_rollback_segments            = 512
nls_date_format                  = YYYY-MM-DD
open_cursors                     = 1024
optimizer_features_enable        = 8.1.7.1
optimizer_index_cost_adj       = 300
optimizer_mode                   = CHOOSE
optimizer_percent_parallel       = 100
parallel_adaptive_multi_user     = TRUE
parallel_broadcast_enabled       = TRUE
parallel_execution_message_size  =8192
parallel_max_servers             = 2000
parallel_min_servers             = 600
parallel_server                 = true
partition_view_enabled           = TRUE
processes                        = 1024
query_rewrite_enabled          = TRUE
sessions                         = 1024
shared_pool_size               = 150000000
sort_area_size                 = 30000000
transaction_auditing             = FALSE
transactions_per_rollback_segment = 1
lm_ress=(70000,70000)
lm_locks=(70000,70000)
log_checkpoints_to_alert=TRUE
replication_dependency_tracking = FALSE
recovery_parallelism=64
large_pool_size =    2598790772
parallel_automatic_tuning = TRUE
```

==========

init_inst1.ora

==========

```
instance_number=1
thread=1
ifile=?/dbs/tkinit.ora
ops_interconnects =
204.152.65.17:204.152.66.33:204.152.66.49:204.152.67.6
5:204.152.67.81
rollback_segments=(rrep11,rrep12,rrep13,rrep14,rrep15,
rrep16,rrep17,rrep18,rrep19,rrep110,rrep111,rrep112,rr
ep113,rrep114,rrep115,rrep116,rrep117,rrep118,rrep119,
rrep120,rrep121,rrep122,rrep123,rrep124,rrep125,rrep12
6,rrep127,rrep128,rrep129,rrep130,rrep131,rrep132,rrep
133,rrep134,rrep135,rrep136,rrep137,rrep138,rrep139,rr
ep140,rrep141,rrep142,rrep143,rrep144,rrep145,rrep146,
rrep147,rrep148,rrep149,rrep150,rrep151,rrep152,rrep15
3,rrep154,rrep155,rrep156,rrep157,rrep158,rrep159,rrep
160,rrep161,rrep162,rrep163,rrep164,rrep165,rrep166,rr
ep167,rrep168,rrep169,rrep170,rrep171,rrep172,rrep173,
rrep174,rrep175,rrep176,rrep177,rrep178,rrep179,rrep18
0,rrep181,rrep182,rrep183,rrep184,rrep185,rrep186,rrep
187,rrep188,rrep189,rrep190,rrep191,rrep192,rrep193,rr
ep194,rrep195,rrep196,rrep197,rrep198,rrep199,rrep1100
,rrep1101,rrep1102,rrep1103,rrep1104,rrep1105,rrep1106
,rrep1107,rrep1108,rrep1109,rrep1110,rrep1111,rrep1112
,rrep1113,rrep1114,rrep1115,rrep1116,rrep1117,rrep1118
,rrep1119,rrep1120,rrep1121,rrep1122,rrep1123,rrep1124
,rrep1125,rrep1126,rrep1127,rrep1128,rrep1129,rrep1130
,rrep1131,rrep1132,rrep1133,rrep1134,rrep1135,rrep1136
,rrep1138,rrep1139,rrep1140,rrep1141,rrep1142,rrep1143
,rrep1144,rrep1145,rrep1146,rrep1147,rrep1148,rrep1149
,rrep1150,rrep1151,rrep1152,rrep1153,rrep1154,rrep1155
,rrep1156,rrep1157,rrep1158,rrep1159,rrep1160,rrep1161
,rrep1162,rrep1163,rrep1164,rrep1165,rrep1166,rrep1167
,rrep1168,rrep1169,rrep1170,rrep1171,rrep1172,rrep1173
,rrep1174,rrep1175,rrep1176,rrep1177,rrep1178,rrep1179
,rrep1180,rrep1181,rrep1182,rrep1183,rrep1184,rrep1185
,rrep1186,rrep1187,rrep1188,rrep1189,rrep1190,rrep1191
,rrep1192,rrep1193,rrep1194,rrep1195,rrep1196,rrep1197
,rrep1198,rrep1199,rrep1200,rrep1201,rrep1202,rrep1203
,rrep1204,rrep1205,rrep1206,rrep1207,rrep1208,rrep1209
,rrep1210,rrep1211,rrep1212,rrep1213,rrep1214,rrep1215
,rrep1216,rrep1217,rrep1218,rrep1219,rrep1220,rrep1221
,rrep1222,rrep1223,rrep1224,rrep1225,rrep1226,rrep1227
,rrep1228,rrep1229,rrep1230,rrep1231,rrep1232,rrep1233
,rrep1234,rrep1235,rrep1236,rrep1237,rrep1238,rrep1239
,rrep1240,rrep1241,rrep1242,rrep1243,rrep1244,rrep1245
,rrep1246,rrep1247,rrep1248,rrep1249,rrep1250,rrep1251
,rrep1252,rrep1253,rrep1254,rrep1255,rrep1256,rrep1257
,rrep1258,rrep1259,rrep1260,rrep1261,rrep1262,rrep1263
,rrep1264,rrep1265,rrep1266,rrep1267,rrep1268,rrep1269
,rrep1270,rrep1271,rrep1272,rrep1273,rrep1274,rrep1275
,rrep1276,rrep1277,rrep1278,rrep1279,rrep1280,rrep1281
,rrep1282,rrep1283,rrep1284,rrep1285,rrep1286,rrep1287
,rrep1288,rrep1289,rrep1290,rrep1291,rrep1292,rrep1293
,rrep1294,rrep1295,rrep1296,rrep1297,rrep1298,rrep1299
,rrep1300,rrep1301,rrep1302,rrep1303,rrep1304,rrep1305
,rrep1306,rrep1307,rrep1308,rrep1309,rrep1310,rrep1311
,rrep1312,rrep1313,rrep1314,rrep1315,rrep1316,rrep1317
,rrep1318,rrep1319,rrep1320,rrep1321,rrep1322,rrep1323
,rrep1324,rrep1325,rrep1326,rrep1327,rrep1328,rrep1329
,rrep1330,rrep1331,rrep1332,rrep1333,rrep1334,rrep1335
,rrep1336,rrep1337,rrep1338,rrep1339,rrep1340,rrep1341
,rrep1342,rrep1343,rrep1344,rrep1345,rrep1346,rrep1347
,rrep1348,rrep1349,rrep1350,rrep1351,rrep1352,rrep1353
,rrep1354,rrep1355,rrep1356,rrep1357,rrep1358,rrep1359
,rrep1360,rrep1361,rrep1362,rrep1363,rrep1364,rrep1365
,rrep1366,rrep1367,rrep1368,rrep1369,rrep1370,rrep1371
,rrep1372,rrep1373,rrep1374,rrep1375,rrep1376,rrep1377
,rrep1378,rrep1379,rrep1380,rrep1381,rrep1382,rrep1383
,rrep1384,rrep1385,rrep1386,rrep1387,rrep1388,rrep1389
,rrep1390,rrep1391,rrep1392,rrep1393,rrep1394,rrep1395
,rrep1396,rrep1397,rrep1398,rrep1399,rrep1400)
instance_groups=groupa, groupb
parallel_instance_group = groupa
```

==========

init_inst2.ora

==========

```
instance_number=2
thread=2
ifile=?/dbs/tkinit.ora
ops_interconnects=204.152.65.18:204.152.66.34:204.152.
66.50:204.152.67.66:204.152.67.82
instance_groups=groupa
parallel_instance_group = groupa
rollback_segments=(rrep21,rrep22,rrep23,rrep24,rrep25,
rrep26,rrep27,rrep28,rrep29,rrep210,rrep211,rrep212,rr
ep213,rrep214,rrep215,rrep216,rrep217,rrep218,rrep219,
rrep220,rrep221,rrep222,rrep223,rrep224,rrep225,rrep22
6,rrep227,rrep228,rrep229,rrep230,rrep231,rrep232,rrep
233,rrep234,rrep235,rrep236,rrep237,rrep238,rrep239,rr
ep240,rrep241,rrep242,rrep243,rrep244,rrep245,rrep246,
rrep247,rrep248,rrep249,rrep250,rrep251,rrep252,rrep25
3,rrep254,rrep255,rrep256,rrep257,rrep258,rrep259,rrep
260,rrep261,rrep262,rrep263,rrep264,rrep265,rrep266,rr
ep267,rrep268,rrep269,rrep270,rrep271,rrep272,rrep273,
rrep274,rrep275,rrep276,rrep277,rrep278,rrep279,rrep28
0,rrep281,rrep282,rrep283,rrep284,rrep285,rrep286,rrep
287,rrep288,rrep289,rrep290,rrep291,rrep292,rrep293,rr
ep294,rrep295,rrep296,rrep297,rrep298,rrep299,rrep2100
,rrep2101,rrep2102,rrep2103,rrep2104,rrep2105,rrep2106
,rrep2107,rrep2108,rrep2109,rrep2110,rrep2111,rrep2112
,rrep2113,rrep2114,rrep2115,rrep2116,rrep2117,rrep2118
,rrep2119,rrep2120,rrep2121,rrep2122,rrep2123,rrep2124
```

```
,rrep2125,rrep2126,rrep2127,rrep2128,rrep2129,rrep2130
,rrep2131,rrep2132,rrep2133,rrep2134,rrep2135,rrep2136
,rrep2138,rrep2139,rrep2140,rrep2141,rrep2142,rrep2143
,rrep2144,rrep2145,rrep2146,rrep2147,rrep2148,rrep2149
,rrep2150,rrep2151,rrep2152,rrep2153,rrep2154,rrep2155
,rrep2156,rrep2157,rrep2158,rrep2159,rrep2160,rrep2161
,rrep2162,rrep2163,rrep2164,rrep2165,rrep2166,rrep2167
,rrep2168,rrep2169,rrep2170,rrep2171,rrep2172,rrep2173
,rrep2174,rrep2175,rrep2176,rrep2177,rrep2178,rrep2179
,rrep2180,rrep2181,rrep2182,rrep2183,rrep2184,rrep2185
,rrep2186,rrep2187,rrep2188,rrep2189,rrep2190,rrep2191
,rrep2192,rrep2193,rrep2194,rrep2195,rrep2196,rrep2197
,rrep2198,rrep2199,rrep2200,rrep2201,rrep2202,rrep2203
,rrep2204,rrep2205,rrep2206,rrep2207,rrep2208,rrep2209
,rrep2210,rrep2211,rrep2212,rrep2213,rrep2214,rrep2215
,rrep2216,rrep2217,rrep2218,rrep2219,rrep2220,rrep2221
,rrep2222,rrep2223,rrep2224,rrep2225,rrep2226,rrep2227
,rrep2228,rrep2229,rrep2230,rrep2231,rrep2232,rrep2233
,rrep2234,rrep2235,rrep2236,rrep2237,rrep2238,rrep2239
,rrep2240,rrep2241,rrep2242,rrep2243,rrep2244,rrep2245
,rrep2246,rrep2247,rrep2248,rrep2249,rrep2250,rrep2251
,rrep2252,rrep2253,rrep2254,rrep2255,rrep2256,rrep2257
,rrep2258,rrep2259,rrep2260,rrep2261,rrep2262,rrep2263
,rrep2264,rrep2265,rrep2266,rrep2267,rrep2268,rrep2269
,rrep2270,rrep2271,rrep2272,rrep2273,rrep2274,rrep2275
,rrep2276,rrep2277,rrep2278,rrep2279,rrep2280,rrep2281
,rrep2282,rrep2283,rrep2284,rrep2285,rrep2286,rrep2287
,rrep2288,rrep2289,rrep2290,rrep2291,rrep2292,rrep2293
,rrep2294,rrep2295,rrep2296,rrep2297,rrep2298,rrep2299
,rrep2300,rrep2301,rrep2302,rrep2303,rrep2304,rrep2305
,rrep2306,rrep2307,rrep2308,rrep2309,rrep2310,rrep2311
,rrep2312,rrep2313,rrep2314,rrep2315,rrep2316,rrep2317
,rrep2318,rrep2319,rrep2320,rrep2321,rrep2322,rrep2323
,rrep2324,rrep2325,rrep2326,rrep2327,rrep2328,rrep2329
,rrep2330,rrep2331,rrep2332,rrep2333,rrep2334,rrep2335
,rrep2336,rrep2337,rrep2338,rrep2339,rrep2340,rrep2341
,rrep2342,rrep2343,rrep2344,rrep2345,rrep2346,rrep2347
,rrep2348,rrep2349,rrep2350,rrep2351,rrep2352,rrep2353
,rrep2354,rrep2355,rrep2356,rrep2357,rrep2358,rrep2359
,rrep2360,rrep2361,rrep2362,rrep2363,rrep2364,rrep2365
,rrep2366,rrep2367,rrep2368,rrep2369,rrep2370,rrep2371
,rrep2372,rrep2373,rrep2374,rrep2375,rrep2376,rrep2377
,rrep2378,rrep2379,rrep2380,rrep2381,rrep2382,rrep2383
,rrep2384,rrep2385,rrep2386,rrep2387,rrep2388,rrep2389
,rrep2390,rrep2391,rrep2392,rrep2393,rrep2394,rrep2395
,rrep2396,rrep2397,rrep2398,rrep2399,rrep2400)
```

Oracle Environment Variables

```
HOME=/export/home/oracle
PATH=/usr/dt/bin:/usr/openwin/bin:.:/export/home/oracle/frame/bin:/export/home/oracle/local/bin:/bin:/usr/bin:/usr/sbin:/etc:/usr/ccs/bin:/export/home/oracle/local/bin:/export/home/oracle/kit/dbgen:/opt/SUNWspro4.2/SUNWspro/bin:/usr/local/bin:/usr/ucb:/usr/local/packages/local/bin:/export/home/oracle/kit/frame/bin:/export/home/oracle/oracle817/bin:/export/home/oracle/frame/bin:/opt/GOODies/bin/:/opt/SUNWscid/bin
LOGNAME=oracle
HZ=100
TZ=US/Pacific
TERM=xterm
SHELL=/usr/bin/tcsh
HOSTTYPE=sun4
VENDOR=sun
OSTYPE=solaris
MACHTYPE=sparc
SHLVL=1
PWD=/audit040201/appendixes/appa
USER=oracle
GROUP=dba
HOST=rep1
REMOTEHOST=minister
MANPATH=/opt/SUNWspro/man:/usr/man:/usr/local/man:/usr/openwin/man:/usr/dt/man:/opt/SUNWmfwm/man
HOST_NAME=rep1
USERNAME=oracle
OPENWINHOME=/usr/openwin
LD_LIBRARY_PATH=/usr/dt/lib:/usr/openwin/lib:/usr/lib:/opt/SUNWspro/lib:/usr/ccs/lib:/vobs/oracle/lib:/export/home/oracle/oracle817/lib:/export/home/oracle/oracle817/rdbms/lib
HELPPATH=/usr/openwin/lib/locale:/usr/openwin/lib/help
XFILESEARCHPATH=/usr/dt/lib/08:40:29N%S:/usr/openwin/lib/locale/%L/%T/%N%S:/usr/openwin/lib/%T/%N%S
MOTIFHOME=/usr/dt
UIDPATH=/opt/SUNWmfwm
XMBINDDIR=/usr/dt/lib/bindings
EDITOR=vi
PRINTER=4op707ap
DISPLAY=rep1:0.0
TEST_FRAME=/export/home/oracle/kit/frame
FRAME_PATH=/export/home/oracle/kit/frame
ENVSET=1
DSS_PATH=/tpch_ff/ff1/300g
DSS_CONFIG=/export/home/oracle/kit/dbgen/
ORACLE_HOME=/export/home/oracle/oracle817
ORACLE_SID=inst1
O=/export/home/oracle/oracle817
KIT_DIR=/export/home/oracle/kit
BUMPX_DIR=/export/home/oracle/kit/bumpx
BUMPX_OUT=/export/home/oracle/kit/bumpx
UTILS=/export/home/oracle/kit/utils
DSS_QUERY=/export/home/oracle/kit/queries
```

Solaris Parameters

(altered from default)

/etc/system

```
*ident  "@(#)system    1.18    97/06/27 SMI" /* SVR4 1.5 */
*
* SYSTEM SPECIFICATION FILE
*

* moddir:
*
*       Set the search path for modules.  This has a format similar to the
*       csh path variable. If the module isn't found in the first directory
*       it tries the second and so on. The default is /kernel /usr/kernel
*
*       Example:
*               moddir: /kernel /usr/kernel /other/modules



* root device and root filesystem configuration:
*
*       The following may be used to override the defaults provided by
*       the boot program:
*
*       rootfs:                 Set the filesystem type of the root.
*
*       rootdev:        Set the root device.  This should be a fully
*                       expanded physical pathname.  The default is the
*                       physical pathname of the device where the boot
*                       program resides.  The physical pathname is
*                       highly platform and configuration dependent.
*
*       Example:
*               rootfs:ufs
*               rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
*
*       (Swap device configuration should be specified in /etc/vfstab.)



* exclude:
*
*       Modules appearing in the moddir path which are NOT to be loaded,
*       even if referenced. Note that 'exclude' accepts either a module name,
*       or a filename which includes the directory.
*
*       Examples:
*               exclude: win
*               exclude: sys/shmsys



* forceload:
```

```
*
*       Cause these modules to be loaded at boot time,
(just before mounting
*       the root filesystem) rather than at first
reference. Note that
*       forceload expects a filename which includes
the directory. Also
*       note that loading a module does not
necessarily imply that it will
*       be installed.
*
*       Example:
*               forceload: drv/foo



* set:
*
*       Set an integer variable in the kernel or a
module to a new value.
*       This facility should be used with caution.
See system(4).
*
*       Examples:
*
*       To set variables in 'unix':
*
*               set nautopush=32
*               set maxusers=40
*
*       To set a variable named 'debug' in the module
named 'test_module'
*
*               set test_module:debug = 0x13


*set rlim_fd_cur = 256
set tune_t_gpgslo=250
set tune_t_minarmem=100
set tune_t_minasmem=250
*set msgsys:msginfo_msgmap=200
set msgsys:msginfo_msgmni=100
*set msgsys:msginfo_msgtql=80
*set msgsys:msginfo_msgseg=2048

* 256 Mbytes ... this is the default
*set shmsys:shminfo_shmmax=268435456
* 2.0 Gbytes ...
*set shmsys:shminfo_shmmax=2147483648
* 3.5 Gbytes ...
*set shmsys:shminfo_shmmax=3758096384
* unlimited ...
set shmsys:shminfo_shmmax=0xffffffffffffffff
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=1024
set shmsys:shminfo_shmseg=500

set semsys:seminfo_semmap=8388608  # set to
semmni*semmsl
set semsys:seminfo_semmni=4096
set semsys:seminfo_semmns=8388608  # set to
semmni*semmsl
set semsys:seminfo_semmnu=4096
set semsys:seminfo_semmsl=2048
set semsys:seminfo_semume=2048
set semsys:seminfo_semopm=100
set semsys:seminfo_semvmx=32767
set msgsys:msginfo_msgmap=2048
set msgsys:msginfo_msgmax=8192
set msgsys:msginfo_msgmnb=16384
*set msgsys:msginfo_msgmni=50
set msgsys:msginfo_msgssz=32
set msgsys:msginfo_msgtql=2048
set msgsys:msginfo_msgseg=32767

set pln:pln_enable_detach_suspend=1
set soc:soc_enable_detach_suspend=1
* pty count
set pt_cnt=1100
set npty=256
set sadcnt=2200
set nautopush=1100


set report_ce_console=1

* tune_t_fsflushr
* DEFAULT: 5
* UNITS: seconds
*4143613  set tune_t_fsflushr=1
set tune_t_fsflushr=1


*  Boost time for autoup.
*  This is the cycle time for a scan of all of memory.
With large memory,
*  the fsflush deamon will get rather busy running
this scan using the default.
* DEFAULT: 30
* UNITS: Seconds
set autoup=600

* Increase descriptor limits

set rlim_fd_max = 2048
set rlim_fd_cur = 1536

* set to allow bigger I/O's
set maxphys=1048576

* vxvm_START (do not remove)
forceload: drv/ses
forceload: drv/vxio
forceload: drv/vxspec
* vxvm_END (do not remove)
* the following 2 lines added 4/18/97 by djm as
suggested by veritas
* the default is 256 but should never go above 65535
set vxio:vol_maxio=8192
set vxio:vol_maxkiocount=32768
* memory alloc chunk size, default 64K
set vxio:voliomem_chunk_size=1048576
* ioctl sizes, set as per A5000 experiments
set vxio:vol_maxioctl=131072
set vxio:vol_maxspecialio=10240
* new 3.0.1 tunable, default 4M, set to 128M.
set vxio:voliomem_maxpool_sz=134217728
* new values set 4/28/00 - djm
* set to allow fast resync's
set vxio:vol_default_iodelay=1
* set to make parallel I/O's stay on the same plex, in
blocks.
set vxio:vol_mvr_maxround=2048
* make recovery do fewer checkpoints (in blocks,
default 20480)
set vxio:vol_checkpt_default=204800
* set by recommendation of Veritas on 6/6/00 for AIO
hangs, djm & gw
set vxio:vol_kmsg_max_send=0x30

* for new 3.0 ge driver
*set ge:ge_intr_mode=0x833

* testing network
*set sq_max_size=0

* turn off ecache scrubber
set ecache_scrub_enable=0
set disable_memscrub=1
```

==========

sunCLUSTER Parameters

(altered from default)

==========

```
Changes to file /etc/opt/SUNWcluster/conf/tpch.cdb:

   cvm.step3_timeout       : 1800
   cvm.step4_timeout       : 1200
   cmm.transition.step8.timeout    : 4000
   cmm.transition.step9.timeout    : 4000
   sma.down_time           : 5000
```

Appendix B. Programs and Scripts

```
========================================
```

load.sh

```
========================================
#!/bin/ksh
echo starting database load
date
echo Starting scuto phase
bumpx.pl −x −o 3tera_scuto.dat −p scuto
echo done scuto
echo Starting dapop phase
echo loading lineitem
bumpx.pl −x −o 3tera_li_dapop.dat −p dapop
echo done loading lineitem
echo loading orders
bumpx.pl −x −o 3tera_or_dapop.dat −p dapop
echo done loading orders
echo loading rest
bumpx.pl −x −o 3tera_papsupcsnr_dapop.dat −p dapop
echo done loading rest
echo creating indexes
bumpx.pl −x −o 3tera_ixcre.dat −p ixcre
echo done creating indexes
echo analyzing
bumpx.pl −x −o 3tera_anlyz.dat −p anlyz
echo done analyzing
echo build last index
bumpx.pl −x −o 3tera_ixcre2.dat −p ixcre
echo done building last index
date
echo all done with database creation
date

SDATE=‘date‘
echo "Setting the random number seed at $SDATE"
PSEED=‘date +%m:%d:%H:%M:%S | sed −e ’s/://g’‘
echo "Using ${PSEED} as seed0"
echo ${PSEED} >
/export/home/oracle/kit/audit/build/seed
cp −p /export/home/oracle/kit/audit/build/seed
/export/home/oracle/kit/audit/seed
echo "Done setting the random number seed at ‘date‘"
>> ${LOG_FILE}

echo running auditors scripts
svrmgrl << !
connect tpcd/tpcd
spool /export/home/oracle/kit/audit/count.out
@/export/home/oracle/kit/audit/count.sql
spool off
spool /export/home/oracle/kit/audit/dbtables.out
@/export/home/oracle/kit/audit/dbtables.sql
exit
!

echo shutting database down

/export/home/oracle/kit/audit/tshut abort
sleep 60
rsh −n rep2 /export/home/oracle/tshut_rep2 abort

================================================
```

bumpx.pl

```
================================================
#!/opt/GOODies/bin/perl
# $Header: bumpx.pl 21−mar−00.16:38:58 mpoess Exp $
#
# bumpx.pl
#
#  Copyright (c) Oracle Corporation 1999, 2000. All
Rights Reserved.
#
#    NAME
#      bumpx.pl − <one−line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file
declares/defines>
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      03/21/00 − fix bug in dapop
controlfile generation for nation/
#    mpoess      03/21/00 − fix ts_data bug for
lineitem & orders
#    mpoess      03/21/00 − fix bug for ops: 2 loaders
per partition
#    mpoess      03/21/00 − make rollback segments
private/public w. option
#    mpoess      01/31/00 − fix bug with undo
#    mpoess      01/21/00 − fix bugs for partitioned
tablespaces
#    mpoess      01/20/00 − add OPS load one partiton
from one node support
#    mpoess      01/07/00 − add support for flatfile
distribution
#    mpoess      01/06/00 − add controlfile generation
performace improvement
#    mpoess      12/16/99 − add partitioning support
for control files
#    mpoess      10/28/99 − adjust bumpx to fixed
dbgen (84 partitioning)
#    mpoess      08/17/99 − change column definition
of base tables
#    mpoess      08/13/99 − add support for reporting
files
#    mpoess      08/08/99 − change file atributes
#    mpoess      07/12/99 − add setpath for dbgen
#    mpoess      07/12/99 − disable −o option for
dbgen
#    mpoess      07/07/99 − fix dbgen phase
#    mpoess      07/06/99 − change path to perl
#    mpoess      07/02/99 − add environment variable
support for conf file
#    mpoess      06/28/99 − add pathname for param.txt
file
#    mpoess      06/28/99 − Copy from TPCD
#    mpoess      06/28/99 − Creation
#    MODIFIED   (MM/DD/YY)
#    mpoess      11/18/98 − add undo rollback segment
support for OPS systems
#    mpoess      11/18/98 − add control log etc.
keywords in SQLLDR call
#    mpoess      11/06/98 − fix bug in print_as_select
#    mpoess      11/05/98 − change name of BUMPC_CTR
variable
#    mpoess      11/04/98 − delete default startup
before index creation
#    mpoess      11/04/98 − adopt Barry Perkins
changes to dbgen
#    mpoess      11/03/98 − change as select handling
#    mpoess      11/02/98 − change background process
handling for NT
#    mpoess      10/26/98 − modifiy dapop so that
multiple partitions can be lo
#    mpoess      10/23/98 − take *waits out from flat
file generation
#    mpoess      10/20/98 − add sdgen, dbgen and dapop
description
#    mpoess      10/12/98 − fix bug in constraint
creation
#    mpoess      10/12/98 − add support for non
partitioned tables
#    mpoess      10/08/98 − add −including new values−
clause in view log defin
#    mpoess      09/27/98 − update links,param2html
#    mpoess      09/25/98 − adding new features
#    akarasik    07/29/98 −
#    akarasik    07/29/98 − Checking in
#     kwong     05/22/98 −  add change storage for
tables and indexes in
#                           chdop
#     kwong      05/21/98 −  fix load pipe
#     kwong      05/21/98 −  add some more default
values in init.ora
#     kwong      05/20/98 −  add parameter nt_port
#     kwong     05/19/98 −  fix control file
generation
#     kwong      05/18/98 −  run utlxplan.sql after
created user
#     kwong      05/18/98 −  add "dbgen" phase
#     kwong      05/15/98 −  calculate partition
values for l_orderkey
#                           and l_partkey
#     kwong      05/13/98 −  add phase "chdop" for
modifying dop as the
#                           end of the build
#     kwong      05/13/98 −  fix the order of create
tablespaces and add
#                           datafiles
#     pswong     04/15/97 −  fix pipeline load
#     pswong     04/02/97 −  analyze partition support
#     pswong     03/27/97 −  introduced ts groups
```

```
#     pswong     03/18/97 -  named pipe loader and
dbgen support
#     pswong     02/28/97 -  more partition support
#     pswong     02/14/97 -  Version 8
#     amozes     10/27/94 -  Creation

#**************************************************
**********************
#**************************************************
**********************
# Main Section
#**************************************************
**********************
#**************************************************
**********************

$os = $ENV{'OS'};
if (($os cmp 'Windows_NT') != 0)
{                               # os is UNIX
    $os = "unix";
    $nt = 0;
    $unix = 1;
}
else
{
    $os = "nt";
    $nt = 1;
    $unix = 0;
}

$| = 1;
$verbose = 0;
$allphases =
"dbcre,shutd,start,dbgen,sccre,scuto,scuvo,sctso,dapop
,ixcre,anlyz,chob,expln,query,sdgen,plcre";
if (($os cmp "unix")==0)
{
    $defphases =
"dbcre,sctso,scuto,dbgen,dapop,anlyz,ixcre";
}
else
{
    $defphases =
"sdgen,shutd,start,dbgen,plcre,dbcre,sctso,scuto,dapop
,scuvo,anlyz,ixcre,chob";
}
$allbmtypes = "tpcd,wisc";
$bmtype = "tpcd" if !defined $bmtype;
$pdfile =  "$ENV{'BUMPX_DIR'}/param.txt";  # This file
contains the description of all possible parameters.
&read_parameter_description;
$runsilent=0;
while ($arg = shift(@ARGV))
{
    if ($arg !~ /-(i|o|t|c|x|p|d|a|s)/)
    {
        $error =  "** Error: Bad argument to $0:
$arg\n";
        &usage;
    }
    $runsilent = 1 if ($arg =~ /-s/);
    $infile = shift(@ARGV) if ($arg =~ /-i/);
    $outfile = shift(@ARGV) if ($arg =~ /-o/);
    $bmtype = shift(@ARGV) if ($arg =~ /-t/);
    $docreate = 1 if (($arg =~ /-c/) || ($arg =~ /-
xc/));
    $doexecute = 1 if (($arg =~ /-x/) || ($arg =~ /-
cx/));
    $phaselist = shift(@ARGV) if ($arg =~ /-p/);
    if ($arg =~ /-d/)
    {
        $defpar = shift(@ARGV);
        &defaults;
        @keys = keys %params;
        while ($#keys >=0)
        {
            $key = pop(@keys);
            if (($defpar cmp "") ==0)
            {
                print $key, "=", $params{$key}, "\n";
            }
            else
            {
                print $key, "=", $params{$key}, "\n" if
($key =~ /$defpar/);
            }
        }
        exit(0);
    }
    if ($arg =~ /-a/)
    {
        $infile = "$ENV{'BUMPX_DIR'}/bumpx.conf" if
!defined $infile;
        $infile = $infile;
        &readfile;
        &defaults;
        @keys = keys %params;
        while ($#keys >=0)
        {
            $key = pop(@keys);
            print $key, "=", $params{$key}, "\n";
        }
        exit(0);
    }
}

$docreate = 0 if !defined $docreate;
$doexecute = 0 if !defined $doexecute;
if (!($docreate || $doexecute))
{
    $error = "** Error: Must specify either -c or -x
or both\n";
    &usage;
}
$infile = "$ENV{'BUMPX_DIR'}/bumpx.conf" if !defined
$infile;
$outfile = "$ENV{'BUMPX_DIR'}/bumpx.dat" if !defined
$outfile;
if ($nt) {
    $listdir = $filedir."list/";
    if (!-e $listfile) {
        system ("mkdir $listdir");
    }
}
if (($os cmp "nt") == 0)
        {
            ## NT Port (Use tmpfile to buffer commands
and nruntpb to synchronize them)
            $tmpfile = "tmp.txt";
            $tmpfile = $filedir.$tmpfile;
            $nruntpb =  "nruntpb.exe";
            ## NT End
        }

if ((!-e $infile) && !$doexecute && $docreate)
{
    $error = "** Error: -i file, $infile, does not
exist\n";
    &usage;
}
if ((!-e $outfile) && !$docreate)
{
    $error = "** Error: -o file, $outfile, does not
exist\n";
    &usage;
}
$phaselist = $defphases if !defined $phaselist;
@phases = split(/,/, $phaselist);

if ($docreate)
{
    open (OUTFILE, ">$outfile") if $docreate;
    &readfile;
    &defaults;
    &docreate;
    close (OUTFILE);
}

## NT Port (Use tmpfile to buffer commands for
nruntpb)
open (TMPFILE, ">$tmpfile") if ( (($os cmp "nt") == 0)
&& $doexecute);
## NT End

&doexecute if $doexecute;

## NT Port
close(TMPFILE) if ( (($os cmp "nt") == 0) &&
$doexecute);
## NT End


exit(0);

sub readfile
{
    $matchon = 0;
    $contin = 0;
    $pkey = "";
    $pval = "";
```

```
    open (INFILE, "$infile");
  WLOOP:
    while ($line = <INFILE>)
    {
        $line = $line."\n" if $line !~ /\n/;
        study $line;
        if ($line =~ /^\*matchon/)
        {
            $matchon = 1;
            next WLOOP;
        }
        if ($line =~ /^\*matchoff/)
        {
            $matchon = 0;
            next WLOOP;
        }
        if ($matchon == 1)
        {
            &dump0($line) if $docreate;
            next WLOOP;
        }
        next WLOOP if $line =~ /^\s*\#/;
        next WLOOP if $line =~ /^\s*\n/;
        if ($contin)
        {
            if ($line =~ /(.*)\&\s*\n/) # still
continuing (changed \ to &)
            {
#                $pval = $pval . $1 . "\n";
                $pval = $pval . $1;
                next WLOOP;
            }
            $line =~ /(.*)\s*\n/; # reached the end
            $pval = $pval . $1;
            $pval =~ s/\?/$ENV{'ORACLE_HOME'}/g;
           &key_exists($pkey);
            if ($result!=1)
            {
                print "Parameter $pkey does not
exist.\nBailing out!\n";
                exit (0);
            }
            $params{$pkey} = $pval;
            $contin = 0;
            $pkey = "";
            $pval = "";
            next WLOOP;
        }
        else
        {
            if ($line =~ /\s*(\S+)\s*=\s*(.*)\&\s*\n/)
            {
                $pkey = $1;
#                $pval = $2 . "\n";
                $pval = $2;
                $contin = 1;
                next WLOOP;
            }
            if ($line =~ /\s*(\S+)\s*=\s*\n/)
            {
                undef($params{$1});
                next WLOOP;
            }
            if ($line !~
/\s*(\S+)\s*=\s*((\S+)|(\S+.*\S+))\s*\n/)
            {
                print "Bad line: $line";
                next WLOOP;
            }
            $tkey = $1;
            $tval = $2;
            $tval =~ s/\?/$ENV{'ORACLE_HOME'}/g;
           if ($tval =~ /\$/) {  # a $ sign at the
beginning of the contents
                                  # of the parameter
value results in a environment
                                   # variable lookup
and substitution
                $tenv = $tval;
                $tval =~ s/\$//g;
                if ($unix) {
                    $tenv =~
s/(.*)\/\$(.*)\/(.*)/\$$2/g;
                }
                $tenvval = `echo $tenv`;
                $tenv =~ s/\$//g;
                chop($tenvval);
                if ($tenvval=~/^\n/) {
                    print "bumpx error: Environment
variable $tval not defined!\nbailing out...\n";
                    exit(1);
                }
                $tval =~ s/$tenv/$tenvval/g;
            }

            &key_exists($tkey);
            if ($result!=1)
            {
                print "Parameter $tkey does not
exists.\nBailing out!\n";
                exit (0);
            }
            $params{$tkey} = $tval;
        }
    }
    close (INFILE);
}

sub docreate
{
    print "Creation pass begun." if $verbose;
    @phases_tmp = @phases; # because I will eat the
elements
    while ($phase=shift(@phases_tmp))
    {
        if ($allphases =~ /$phase/)
        {
            $nextphase = shift(@phases_tmp);
            unshift (@phases_tmp,$nextphase) if
($nextphase);
            print "\n  Creating phase \'$phase\'" if
$verbose;
            &dump0("%b-$phase\n");
            &prepmulti();
            &dump0("*bgon=$params{$phase.'_max_bg'}");
            eval (&$phase);
            &dump0("*wait");
            &dump0("*bgoff");
            &dump0("%e-$phase\n");
        }
        else
        {
            print "\n  Phase \'$phase\' not built
in...assuming a \*match block being used";
        }
    }
    print "\nCreation pass complete.\n" if $verbose;
}

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

```
    }
    close (INFILE);

# Then, do all of the requested phases
    $execctr = 0;
    foreach $phase (@phases)
    {
        $phase_cmd_num = 0;
        print "\n  Executing phase \'$phase\'" if
$verbose;
        $bg = 0;
        open (INFILE, $outfile);
      WLOOP2:
        while ($line = <INFILE>)
        {
            study $line;
            next WLOOP2 if $line =~ /^\s*\#/;
            next WLOOP2 if $line =~ /^\s*\n/;
            if ($line =~ /^\*ignon/)
            {
                $ignon = 1;
                next WLOOP2;
            }
            if ($line =~ /^\*ignoff/)
            {
                $ignon = 0;
                next WLOOP2;
            }
            next WLOOP2 if ($ignon == 1);
            if ($line =~ /^%b-$phase/)
            {
                $insection = 1;
                $execcmd = "";
                next WLOOP2;
            }
            next WLOOP2 if ($insection != 1);
            if ($line =~ /^%e-$phase/)
            {
                $insection = 0;
                &execute ($execcmd);
                last WLOOP2;
            }
            if ($line =~ /^\*(.*)/)
            {
                &execute ($execcmd);
                if (($1 =~ /bgo/) || ($1 =~ /wait/) ||
($1 =~ /ignore/))
                {
                    $execcmd = $line;
                    next WLOOP2;
                }
                $line =~ /^(\*.*\S+)\s*\n$/;
                $execcmd = $commands{$1};
                next WLOOP2;
            }
            if ($line =~ /^\{(.*)\}/)
            {
                $insert = "";
                $insert = $1;
                $execcmd =~ s/\{\}/$insert/;
                next WLOOP2;
            }
            if ($line =~ /^\{(.*)$/)
            {
                $insubsection = 1;
                $insert = "";
                $insert = $1;
                next WLOOP2;
            }
            if ($line =~ /^(.*)\}/)
            {
                $insubsection = 0;
                $insert = $insert . $1;
## NT Port (Ignore '\n')
               if (($os cmp "nt") == 0)
               {
                   $insert =~ /(.*)\n$/s;
                   $insert = $1;
               }
## NT End
                $execcmd =~ s/\{\}/$insert/;
                next WLOOP2;
            }
            $insert = $insert . $line if
($insubsection == 1);
        }
        close (INFILE);
    }
    print "\nExecution pass complete.\n" if $verbose;
}

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

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

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

        if ($cmd =~ /(s|g)etenv/)
        {
            @lines = split(/\n/, $cmd);
            $cmd = "";
```

```
            foreach $line (@lines)
            {
                while (1)
                {
                    last if ($line !~ /getenv/);
                    $line =~
/(.*)\*getenv\(([^(\)|\*)]*)\)(.*)/;
                    $line = $1 . $ENV{$2} . $3;
                }
                if ($line =~ /toto/) #we do not want
to use this for now
                {
                    $line =~ /setenv\s+(\S+)\s+(\S+)/;
                    $ENV{$1} = $2;
                }
                else
                {
                    $cmd = $cmd . $line. "\n";
                }
            }
        }
        return if ($cmd !~ /\S+/); # return if nothing
left to execute

        $execctr++;
        $ENV{'BUMPX_CTR'} = $$.'-'.$execctr;

        if (($os cmp "unix") == 0)
        {
            if ($bg == 1)
            {
                print "." if $verbose;
                $fpid = fork;
                if ($fpid == 0)
                {
                    exec ($cmd);
                    print "exec\'d command
failed:\n$cmd\nreason: $!\n";
                    exit(-1);
                }
                unshift (@wpids, $fpid);
                $bgrun = $bgrun + 1;
                &execute ("*wait") if (($bgrun >=
$bgmax) && ($bgmax >= 0));
            }
            else
            {
                system ($cmd);
                print "system\'d command
failed:\n$cmd\nreason: $? ($!)\n" if $?;
            }
        }
        else ## NT support
        {
         ## NT Port (Submit background tasks if there
are bgmax of them, otherwise write to tmpfile)
            if ($bg == 1)
            {
                print "." if $verbose;
                if ($bgrun < $bgmax)
                {
                    $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                    ++$phase_cmd_num;
                    print TMPFILE $cmd;
                    $bgrun = $bgrun + 1;
                }
                else
                {
                    close(TMPFILE);
                    system("cat $tmpfile >>
$listdir$phase.lst");
                    system("$nruntpb -p < $tmpfile");
                    if ($?) {
                        print "system command
failed:\n$nruntpb < $tmpfile\nreason: $? ($!)\n";
                        print "Please check the
contents in the input file.\n";
                        exit(-1);
                    }
                    open(TMPFILE, ">$tmpfile");
                    $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                    ++$phase_cmd_num;
                    print TMPFILE $cmd;
                    $bgrun = 1;
                }
            }
            else
            {
                $cmd =~
s/phase\#.lst/$listdir$phase\_$phase_cmd_num.lst/;
                ++$phase_cmd_num;
                print TMPFILE $cmd;
                close(TMPFILE);
                system("cat $tmpfile >>
$listdir$phase.lst");
                system ("sh $tmpfile");
                if ($?) {
                    print "system\'d command
failed:\nsh $tmpfile\nreason: $? ($!)\n";
                    print "Please check the contents in
the shell script.\n";
                    exit(-1);
                }
                open(TMPFILE, ">$tmpfile");
            }
        }  ## NT support End
    }
}

sub usage
{
    print "Usage:\n";
    print " $0 [-c] [-x] [-i infile] [-o outfile] [-p
phaselist] [-t type]\n";
    print "   -c : create intermediary file (needed
for execution)\n";
    print "   -x : execute intermediary file\n";
    print "   -i : configuration file to be used\n";
    print "        defaults to bumpx.conf in
\$BUMPX_DIR or \$CWD\n";
    print "   -o : intermediary file to be created
and/or used\n";
    print "        defaults to bumpx.dat in
\$BUMPX_DIR or \$CWD\n";
    print "   -p : list of phases to
create/execute\n";
    print "        phaselist is a comma separated list
of phases in order\n";
    print "        possible phases are:\n";
    print "          sdgen = seed file generation\n";
    print "          dbgen = data flat file
generation\n";
    print "          plcre = NT raw partition and
links creation\n";
    print "          dbcre = database creation\n";
    print "          shutd = shutdown database (on all
instances)\n";
    print "          start = startup database (on all
instances)\n";
    print "          sccre = schema creation\n";
    print "          sctso = schema creation
(tablespaces only)\n";
    print "          scuto = schema creation (user and
tables only)\n";
    print "          scuvo = schema creation (views
only)\n";
    print "          dapop = data population\n";
    print "          ixcre = index creation (including
constraints)\n";
    print "          anlyz = analyze objects\n";
    print "          chob  = change parameters of
objects\n";
    print "          expln = create explain plans\n";
    print "          query = run and time queries\n";
    print "        defaults to $defphases\n";
    print "   -t : type of benchmark\n";
    print "        enables benchmark-specific
defaults\n";
    print "        current possibilties are:
$allbmtypes\n";
    print "        defaults to tpcd\n";
    print "   -s : run silent (no parameter checking
is done)\n";
    print "\n";
    print "Examples:\n";
    print " $0 -c -o file.out\n";
    print "   This will create an intermediary file
named file.out for the default\n";
    print "   phases using bumpx.conf which can be
executed in the future.\n";
    print " $0 -x -p dapop\n";
    print "   Executes data population phase of
intermediary file bumpx.dat.\n";
    print " $0 -cx -p dbcre,dapop\n";
    print "   This will create an intermediary file,
bumpx.dat, using configuration\n";
    print "   file, bumpx.conf, for both the database
creation phase and the data\n";
```

```
    print "   population phase, and then execute that
file.\n";
    print "\n";
    print "$error\n";
    exit(-1);
}

#****************************************************
**********************
#****************************************************
**********************
# All Known Phases
#****************************************************
**********************
#****************************************************
**********************

sub dbcre
{

&dump0("################################################
##############################");
    &dump0("# Database Creation Phase");
    &time0("Begin database creation");

# Shut down anything that may have been running
    &shutdb();

# Create database (system tablespace/datafile, redo
logs, rollback segments)
    @sys_files = split (/,/,
$params{'ts_sys_datafiles'});
    $sys_file = shift(@sys_files);
    @log_files = split (/,/,
$params{'ts_log_datafiles'});
    &dump0("# creating database and initial rollback
segment");
    $cmd .= "startup pfile=
$params{'startupfile_dbcre'} nomount;\n";
    $cmd .= "create database\n";
    $cmd .= "  controlfile reuse\n" if
!($params{'db_ctlreuse'} =~ /false/);
    for ($i = 0; $i < $params{'ts_log_#files_pt'};
$i++)
    {
        ($i == 0) ? ($addname = "logfile") : ($addname
= "        ");
        (($i+1) == $params{'ts_log_#files_pt'}) ?
($addcomma = "") : ($addcomma = ",");
        $log_file = shift(@log_files);
        if (($os cmp "unix") == 0)
        {
            $cmd .= "  $addname
'$params{'ts_log_area'}$log_file' size
$params{'ts_log_size'}
$params{'ts_log_options'}$addcomma\n";
        }
        else
        {
            $cmd .= "  $addname (
'$params{'ts_log_area'}$log_file') size
$params{'ts_log_size'}
$params{'ts_log_options'}$addcomma\n";
        }
    }
    $cmd .= "  maxlogfiles
$params{'db_maxlogfiles'}\n" if defined
$params{'db_maxlogfiles'};
    $cmd .= "  maxlogmembers
$params{'db_maxlogmembers'}\n" if defined
$params{'db_maxlogmembers'};
    $cmd .= "  maxloghistory
$params{'db_maxloghistory'}\n" if defined
$params{'db_maxloghistory'};
    $cmd .= "  datafile
'$params{'ts_sys_area'}$sys_file' size
$params{'ts_sys_size'} $params{'ts_sys_options'}\n";
    $cmd .= "  maxdatafiles
$params{'db_maxdatafiles'}\n" if defined
$params{'db_maxdatafiles'};
    $cmd .= "  maxinstances
$params{'db_maxinstances'}\n" if defined
$params{'db_maxinstances'};
    $cmd .= "  maxarchlogs
$params{'db_maxarchlogs'}\n" if defined
$params{'db_maxarchlogs'};
    $cmd .= "  archivelog\n" if defined
$params{'db_archivelog'};
    $cmd .= "  character set $params{'db_charset'}\n"
if defined $params{'db_charset'};
    $cmd .= ";\n";
    $cmd .= "\n";
    $cmd .= "create $params{'ts_undo_rs_type'}
rollback segment t_rs1 ";
    $cmd .= "  storage $params{'ts_undo_rs_storage'}"
if defined $params{'ts_undo_rs_storage'};
    $cmd .= ";\n";
    $cmd .= "\n";
    $cmd .= "alter rollback segment t_rs1 online;\n";
    $cmd .= "\n";
    $cmd .= "shutdown\n"; #^^sd
    &dump1("*sql");
    &dump0("*wait");

# This startup is in its own session because of some
weird bug I've been
# seeing on the SP2; otherwise, it's in the previous
session
#^^su
#    $cmd = "";
#    $cmd .= "startup pfile=$iofname[1];\n";
#    &dump1("*sql");
#    &dump0("*wait");
    startdb("dbcre");
    &dump0("*wait");
    @ops_nodes =  split(/,/,$params{'ops_nodes'});
    if (defined $params{'ops_nodes'}) {
       foreach $ops_node (@ops_nodes)
       {
           if ($ops_node =~ /undo/)
           {
              $ts_undo = 'undo';
           }
           else {
              $ts_undo = 'undo_'.$ops_node;
           }
           if ($params{'skip_ts'} !~ /$ts_undo/)
           {
              &add_ts_rollb ($ts_undo);
           }
       }
    } else {
       $ts_undo = 'undo';
    }


    # currently set up for the foreground - maybe
should be changed
    &dump0("# creating extra logfile threads");
    for ($i = 1; $i < $params{'ts_log_#threads'};
$i++)
    {
        $thrno = $i + 1;
        $cmd .= "alter database add logfile thread
$thrno\n";
        for ($j = 0; $j < $params{'ts_log_#files_pt'};
$j++)
        {
            (($j+1) == $params{'ts_log_#files_pt'}) ?
($addcomma = ";") : ($addcomma = ",");
            $log_file = shift(@log_files);
            $cmd .= "
'$params{'ts_log_area'}$log_file' size
$params{'ts_log_size'}
$params{'ts_log_options'}$addcomma\n";
        }
        $cmd .= "alter database enable public thread
$thrno;\n";
    }
    &dump1("*sql");
    &dump0("*wait");

# Build data dictionary
    &dump0("# building data dictionary");
    $cmd .= "set termout off\n";
    $cmd .= "set echo off\n";
    $cmd .= '@' . "$params{'dd_sql_area'}" .
"catalog.sql;\n";
    $cmd .= '@' . "$params{'dd_sql_area'}" .
"catparr.sql;\n";
    $cmd .= '@' . "$params{'dd_sql_area'}" .
"catproc.sql;\n";
    $cmd .= "connect system/manager\n";
    $cmd .= '@' . "$params{'dd_sql_area'}" .
"utlxplan.sql;\n";
    $cmd .= '@' . "$params{'dd_sqlplus_area'}" .
"pupbld.sql;\n";
    &dump1("*sql");
    &dump0("*wait");
    &time0("Done database creation");
```

```
# prepare for multi-user
#^^sd
    $cmd .= "shutdown;\n";
    &dump1("*sql");
    &dump0("*wait");
    &startdb("dbcre");
}                               # end dbcre

sub shutd
{

&dump0("############################################
##############################");
    &dump0("# Shutdown Database - All Instances");
    &shutdb();
    &time0("Done database shutdown - all instances");
}

sub start
{

&dump0("############################################
##############################");
    &dump0("# Startup Database - All Instances");
    &startdb($nextphase);
    &time0("Done database startup - all instance");
}

sub sdgen
{

&dump0("############################################
##############################");
    &dump0("# Data (Seed File) Generation Phase");
    &time0("Begin creating seed files");
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        $bgopt = "";
        $dbgprc  =
$params{'load_dbgen_'.$table.'_option_C'};
        if (! $seen{$dbgprc}++) {
                $dbgopt = $verbose ? "-v " : "";
                $dbgopt  .= "-O s -s " .
$params{'scale_factor'};
                $cmd .= $dbgopt . " -C " . $dbgprc;
                $cmd .= sprintf(" 2>>
dbgen_seed_%d.log", $dbgprc) if $log_output;
                $cmd .= "\n";
                &dump1("*dbgen");
                &time0("Done creating seed files for
degree $dbgprc");
        }
    }
    &dump0("*wait");
    &time0("Done creating seed files");
    &dump0("*wait");
} # end of sdgen

sub dbgen
{

&dump0("############################################
##############################");
    &dump0("# Data (Flat File) Generation Phase");
    &time0("Begin creating flat files");

# change DSS_PATH
    $cmd = "setenv DSS_PATH
$params{'load_flatfile_area'}";
    &dump1("*sh");

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        $curts = $params{'tab_'.$table.'_ts'};
        $usels = 0;
        $lfexist = 0;
        $bfexist = 0;
        $dafexist = 0;
        $difexist = 0;
        $ffexist = 0;

        # see if we are using pipes
        $up = $params{'load_use_pipes'} =~ /$table/ ?
1 : 0;

        $dp = $params{'tab_'.$table.'_load_degpar'};
        $dgp = ($params{'load_dbgen_partition'} =~
/$table/) ? 1 : 0;
        $dbgpar =
$params{'load_dbgen_'.$table.'_option_C'};

# create all of the "executable" load sections
# These variables will hold for all loaders for a
given table
        $upwd = "";
        $parbool = "";
        $dirbool = "";
        $silent = "";
        $dismax = "";
        $file = "";
        $errors = "";
        $rows = "";
        $bsize = "";
        $load = "";
        $skip = "";
        $dbgopt = "";
        $dbgtab = "";

        if ($table =~ /lineitem/)
        {
            $dbgtab = "L";
        }
        elsif ($table =~ /orders/)
        {
            $dbgtab = "O";
        }
        elsif ($table =~ /partsupp/)
        {
            $dbgtab = "S";
        }
        elsif ($table =~ /part/)
        {
            $dbgtab = "P";
        }
        elsif ($table =~ /customer/)
        {
            $dbgtab = "c";
        }
        elsif ($table =~ /supplier/)
        {
            $dbgtab = "s";
        }
        elsif ($table =~ /nation/)
        {
            $dbgtab = "n";
        }
        elsif ($table =~ /region/)
        {
            $dbgtab = "r";
        }

        $dbgopt .= "-T ".$dbgtab." ";
        $dbgopt .= "-s ". $params{'scale_factor'} . "
";
        $dbgopt .= "-C ";
        $dbgopt .=
$params{'load_dbgen_'.$table.'_option_C'};
        # if using pipes, create the pipes

        $nperset = 1;

        if ($dgp && $dp > 1)
        {
            $nset = ($params{'tab_'.$table.'_#part'} >
$dp) ? 1
                : int($dp /
$params{'tab_'.$table.'_#part'});
            $nperset = int($dbgpar / $nset);
        }

        for ($i=0;$i<$dbgpar;$i++)
        {
            # create script to generate flat files for
each dbgpar
            $cmd = sprintf("%s -S %d", $dbgopt, ($i +
1));
            $fpre =
$params{'load_flatfile_area'}.$params{'load_dbgen_'.$t
able.'_output_prefix'};
```

```
            if ($dgp)
            {
                # create script to generate partitioned
flat files

                # if the output prefix name is
specified with #
                # divide output file names into
$dp/#numpart sets
                if ($fpre =~ /\#/)
                {
                    $snum = 1;
                    $snum = (int($i/$nperset) + 1) if
($dp > 1);
                    $fpre =~ s/\#/$snum/g;
                }
                # I commented this out to enable
support for fixed dbgen (fixed to 84 partitions)
                $cmd .= " -p -I ";
                $cmd .=
$params{'load_dbgen_'.$table.'_input_params'};
                $cmd .= sprintf(" -o %s_%d ", $fpre,($i
+ 1));
                # the folowing is the new stuff
                #$cmd = "-D ".$cmd;

            } else {
                # $cmd .= sprintf(" -o $fpre"); this is
only temporary
            }
            $cmd = $verbose ? "-v -f " . $cmd : "-f "
. $cmd;
#           $cmd .= " & ";
            $cmd .= "\n";
            &dump1("*dbgen");
        }
        &dump0("*wait");
        &time0("Done creating flat files for table
$table");
    }
    &dump0("*wait");
    &time0("Done creating flat files");
} # end of dbgen


sub dbgen_old
{

&dump0("##############################################
##############################");
    &dump0("# Data (Flat File) Generation Phase");
    &time0("Begin creating flat files");

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        $dbgprc =
$params{'load_dbgen_'.$table.'_option_C'};
        $curts = $params{'tab_'.$table.'_ts'};
        $dp = $params{'tab_'.$table.'_load_degpar'};
        $usels = 0;
        $lfexist = 0;
        $bfexist = 0;
        $dafexist = 0;
        $difexist = 0;
        $ffexist = 0;

        # see if we are using pipes
        $up = $params{'load_use_pipes'} =~ /$table/ ?
1 : 0;

        $dgp = ($params{'load_dbgen_partition'} =~
/$table/) ? 1 : 0;
        $dbgpar =
$params{'load_dbgen_'.$table.'_option_C'};

# create all of the "executable" load sections
# These variables will hold for all loaders for a
given table
        $upwd = "";
        $parbool = "";
        $dirbool = "";
        $silent = "";
        $dismax = "";
        $file = "";
        $errors = "";
        $rows = "";
        $bsize = "";
        $load = "";
        $skip = "";
        $dbgopt = "";
        $dbgtab = "";

        if ($table =~ /lineitem/)
        {
            $dbgtab = "L";
        }
        elsif ($table =~ /orders/)
        {
            $dbgtab = "O";
        }
        elsif ($table =~ /partsupp/)
        {
            $dbgtab = "S";
        }
        elsif ($table =~ /part/)
        {
            $dbgtab = "P";
        }
        elsif ($table =~ /customer/)
        {
            $dbgtab = "c";
        }
        elsif ($table =~ /supplier/)
        {
            $dbgtab = "s";
        }
        elsif ($table =~ /nation/)
        {
            $dbgtab = "n";
        }
        elsif ($table =~ /region/)
        {
            $dbgtab = "r";
        }

        $dbgopt .= "-T ".$dbgtab." ";
        $dbgopt .= "-s ". $params{'scale_factor'} . "
";
        $dbgopt .= "-C ";
        $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?
            $params{'load_dbgen_'.$table.'_option_C'}
:
            $params{'load_dbgen_def_option_C'});
        if ($dgp)
        {
            $dbgopt .= " -p -i ";
            $dbgopt .= (defined
$params{'load_dbgen_'.$table.'_input_params'}) ?
$params{'load_dbgen_'.$table.'_input_params'} :
$params{'load_dbgen_def_input_params'};
        }

        # Start - Barry N. Perkins 8/27/1998
#
        # included the following lines to copy flat
files#
        # from default location to backup location
#
        # Defined a new variable for backup location
#
        $fbkp = (defined
$params{'load_dbgen_flatfile_backup'}) ?
            $params{'load_dbgen_flatfile_backup'} :
            0;
        $bpre = (defined
$params{'load_dbgen_backup_'.$table.'_prefix'}) ?

$params{'load_dbgen_backup_'.$table.'_prefix'} :
            $params{'load_dbgen_backup_area'} .
$table;
        # End   - Barry N. Perkins 8/27/1998
#
        # if using pipes, create the pipes
        if($dgp)
        {
            # create script to generate partitioned
flat files for each dbgpar

            $nperset = 1;

            if ($dgp && $dp > 1)
```

```
            {
                $nset =
($params{'tab_'.$table.'_#part'} > $dp) ? 1 : int($dp
/ $params{'tab_'.$table.'_#part'});
                $nperset = int($dbgpar / $nset);
            }

            for ($i=0;$i<$dbgpar;$i++)
            {
                $cmd = sprintf("%s -S %d", $dbgopt, ($i
+ 1));
                if ($dgp)
                {
                    # if the output prefix name is
specified with #
                    # divide output file names into
$dp/#numpart sets
                    $fpre = (defined
$params{'load_dbgen_'.$table.'_output_prefix'}) ?
                        $params{'load_dbgen_'.$table.'_
output_prefix'} :
                        $params{'load_flatfile_area'} .
$table;
                    if ($fpre =~ /\#/)
                    {
                        $snum = 1;
                        $snum = (int($i/$nperset) + 1)
if ($dp > 1);
                        $fpre =~ s/\#/$snum/g;
                    }
                    $cmd .= sprintf(" -o %s_%d ",
$fpre,($i + 1));
                }
                $cmd = $verbose ? "-v -f " . $cmd : "-f
" . $cmd;
                $cmd .= sprintf(" 2>> %s_%d.log",
$table, ($i + 1)) if $log_output;
#               $cmd .= " & ";
                $cmd .= "\n";
                &dump1("*dbgen");
            }
#           &dump0("*wait");
        }
        else
        {
            $cmd = $verbose ? "-v " : "";
            $cmd .= "-f " . $dbgopt;
            $cmd .= sprintf(" 2>> %s.log", $table) if
$verbose;
            $cmd .= "\n";
            &dump1("*dbgen");
        }
        # Start - Barry N. Perkins 8/27/1998
#
        # Included the following lines to copy flat
files#
        # from default location to backup location
#
        # Copy from default location to backup
location  #
        if ($fbkp) {
            $cmd = sprintf("copy %s%s*.* %s*.*\n",
$fpre, $table, $bpre);
            &dump1("*sh");
            &dump0("*wait");
        }
        # End   - Barry N. Perkins 8/27/1998
#
        &time0("Done creating flat files for table
$table");
    }
    &dump0("*wait");
    &time0("Done creating flat files");
} # end of dbgen

sub dbgen_my_split_version # I will delete this and
dbgen_nt and dbgen_unix as soon as new dbgen is fully
tested
{
    if (($os cmp "unix") ==0)
    {
        &dbgen_unix();
    }
    else
    {
        &dbgen_nt();
    }
}

sub dbgen_nt
{

&dump0("#########################################
##############################");
    &dump0("# Data (Flat File) Generation Phase");
    &time0("Begin creating flat files");

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        $curts = $params{'tab_'.$table.'_ts'};
        $dp = $params{'tab_'.$table.'_load_degpar'};
        $usels = 0;
        $lfexist = 0;
        $bfexist = 0;
        $dafexist = 0;
        $difexist = 0;
        $ffexist = 0;
        $up = 0;
        $dgp = 0;
        $dbgpar = 1;

        # see if we are using pipes

        $up = 1 if ($params{'load_use_pipes'} =~
/$table/);
        $dgp = 1 if ($params{'load_dbgen_partition'}
=~ /$table/);
        $dbgpar = $params{'load_dbgen_def_option_C'}
if defined $params{'load_dbgen_def_option_C'};
        $dbgpar =
$params{'load_dbgen_'.$table.'_option_C'} if defined
$params{'load_dbgen_'.$table.'_option_C'};


# create all of the "executable" load sections
# These variables will hold for all loaders for a
given table
        $upwd = "";
        $parbool = "";
        $dirbool = "";
        $silent = "";
        $dismax = "";
        $file = "";
        $errors = "";
        $rows = "";
        $bsize = "";
        $load = "";
        $skip = "";
        $dbgopt = "";
        $dbgtab = "";


            if ($table =~ /lineitem/)
            {
                $dbgtab = "L";
            }
            elsif ($table =~ /orders/)
            {
                $dbgtab = "O";
            }
            elsif ($table =~ /partsupp/)
            {
                $dbgtab = "S";
            }
            elsif ($table =~ /part/)
            {
                $dbgtab = "P";
            }
            elsif ($table =~ /customer/)
            {
                $dbgtab = "c";
            }
            elsif ($table =~ /supplier/)
            {
                $dbgtab = "s";
            }
            elsif ($table =~ /nation/)
            {
                $dbgtab = "n";
            }
            elsif ($table =~ /region/)
            {
```

```
            $dbgtab = "r";
        }

    if ($up) {
        $dbgopt .= "-T ".$dbgtab." ";
        $dbgopt .= "-s ". $params{'scale_factor'}
. " ";
        $dbgopt .= "-C ";
        $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?
                    $params{'load_dbgen_'.$table.'_
option_C'} :
                    $params{'load_dbgen_def_option_
C'});
        if ($dgp) {
            $dbgopt .= " -p -i ";
            $dbgopt .= (defined
$params{'load_dbgen_'.$table.'_input_params'}) ?
$params{'load_dbgen_'.$table.'_input_params'} :
$params{'load_dbgen_def_input_params'};
        }
    }
    else
    {
        $dbgopt .= "-T ".$dbgtab." ";
        $dbgopt .= "-s ". $params{'scale_factor'}
. " ";
        $dbgopt .= "-C ";
        $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?
                    $params{'load_dbgen_'.$table.'_
option_C'} :
                    $params{'load_dbgen_def_option_
C'});
    }

    # if using pipes, create the pipes

    if($up)
    {
        # create script to generate partitioned
flat files for each dbgpar

        $nperset = 1;

        if ($dgp && $dp > 1) {
            $nset = int($dp /
$params{'tab_'.$table.'_#part'});
            $nperset = int($dbgpar / $nset);
        }

        for ($i=0;$i<$dbgpar;$i++) {
            $cmd = sprintf("%s -S %d", $dbgopt, ($i
+ 1));
            if ($dgp) {
                # if the output prefix name is
specified with #
                # divide output file names into
$dp/#numpart sets
                $fpre = (defined
$params{'load_dbgen_'. $table.'_output_prefix'}) ?
$params{'load_dbgen_'.$table.'_output_prefix'} :
$params{'load_dbgen_def_output_prefix'};
                if ($fpre =~ /\#/) {
                    $snum = 1;
                    $snum = (int($i/$nperset) + 1)
if ($dp > 1);
                    $fpre =~ s/\#/$snum/g;
                }
                $cmd .= sprintf(" -o %s_%d ",
$fpre,($i + 1));
            }

            $cmd = " -f " . $cmd;
#           $cmd .= " & ";

            $cmd .= "\n";
            &dump1("*dbgen");
        }

        &dump0("*wait");

         # create script to concatenate partitioned
flat files for each dbgpar into partitioned flat files
by partitions and number of partitioned files in each
partition

         if ($dgp) {
            $nptn =
$params{'tab_'.$table.'_#part'};
            for ($i=0;$i<$dbgpar;$i++) {
                # if the output prefix name is
specified with #
                # divide output file names into
$dp/#numpart sets
                $fpre = (defined
$params{'load_dbgen_'. $table.'_output_prefix'}) ?
$params{'load_dbgen_'.$table.'_output_prefix'} :
$params{'load_dbgen_def_output_prefix'};
                if ($fpre =~ /\#/) {
                    $snum = 1;
                    $snum = (int($i/$nperset) + 1)
if ($dp > 1);
                    $fpre =~ s/\#/$snum/g;
                }
                $pnum = ($i % $nptn) + 1;
                $cmd = sprintf("copy %s_*.%d
%s.%d\n", $fpre, $pnum, $fpre, $pnum);
                &dump1("*sh");
                &dump0("*wait");
                $cmd = sprintf("del %s_*.%d\n",
$fpre, $pnum);
                &dump1("*sh");
                &dump0("*wait");
            }
        }
        else
        {
            $cmd = "-f " . $dbgopt;
            $cmd .= "\n";
            &dump1("*dbgen");
        }
        &dump0("*wait");
        &time0("Done creating flat files for table
$table");
    }
    &dump0("*wait");
    &time0("Done creating flat files");
} # dbgen_nt


sub dbgen_unix
{

&dump0("###############################################
################################");
    &dump0("# Database Population Phase");

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        $curts = $params{'tab_'.$table.'_ts'};
        $dp = $params{'tab_'.$table.'_load_degpar'};
        $usels = 0;
        $lfexist = 0;
        $bfexist = 0;
        $dafexist = 0;
        $difexist = 0;
        $ffexist = 0;
        $up = 0;
        $dgp = 0;
        $dbgpar = 1;

        # see if we are using pipes

        $up = 1 if ($params{'load_use_pipes'} =~
/$table/);
# <bug> parts/partsupp can be confused
        $dgp = 1 if ($params{'load_dbgen_partition'}
=~ /$table/);
        $dbgpar = $params{'load_dbgen_def_option_C'}
if defined $params{'load_dbgen_def_option_C'};
        $dbgpar =
$params{'load_dbgen_'.$table.'_option_C'} if defined
$params{'load_dbgen_'.$table.'_option_C'};

         $params{'tab_'.$table.'_load_ctlf'} =
$table.'.ctl' if !defined
$params{'tab_'.$table.'_load_ctlf'};
         @ctlf = split(/,/,
$params{'tab_'.$table.'_load_ctlf'});
         $err = "ctl" if ((@ctlf > 1) && (@ctlf !=
```

```
$dp));
        @logf = split(/,/,
$params{'tab_'.$table.'_load_logf'});
        $err = "log" if ((@logf > 1) && (@logf !=
$dp));
        $lfexist = 1 if @logf > 0;
        @badf = split(/,/,
$params{'tab_'.$table.'_load_badf'});
        $err = "bad" if ((@badf > 1) && (@badf !=
$dp));
        $bfexist = 1 if @badf > 0;
        @datf = split(/,/,
$params{'tab_'.$table.'_load_datf'});
        $err = "dat" if ((@datf > 1) && (@datf !=
$dp));
        $dafexist = 1 if @datf > 0;
        @disf = split(/,/,
$params{'tab_'.$table.'_load_disf'});
        $err = "dis" if ((@disf > 1) && (@disf !=
$dp));
        $difexist = 1 if @diff > 0;
       @filf = ();
#        $params{'tab_'.$table.'_load_filf'} =
$params{$curts.'_datafiles'} if
($params{'tab_'.$table.'_load_filf'} =~
/alltsdatafiles/);
        if ($params{'tab_'.$table.'_load_filf'} =~
/alltsdatafiles/)
        {
          $params{'tab_'.$table.'_load_filf'} =
$params{$curts.'_datafiles'}
        }
        @filf = split(/,/,
$params{'tab_'.$table.'_load_filf'}) if defined
$params{'tab_'.$table.'_load_filf'};
       # relax the requirement a little bit by
allowing round robining
       $numfil = @filf;
#       if (($numfil > 1) && ($numfil < $dp) && (($dp
% $numfil) == 0)) {
       if (($numfil > 1) && ($numfil < $dp)) {
#           $p = $dp / $numfil;
            $fil = "";
            for ($i = 0; $i < $dp; $i++)
            {
                if ($i == 0) {
                    $fil =
$params{'tab_'.$table.'_load_filf'};
                } else {
                    $fil =
join(',',$fil,$params{'tab_'.$table.'_load_filf'});
                }
            }
            @filf = split(/,/, $fil);
        }
        $err = "fil" if ((@filf > 1) && ($dp !=
@filf));
        $ffexist = 1 if @filf > 0;
        if ($err)
        {
            print "Error with load parameters for
table $table\n";
            print "  degree is $dp, $err list has bad
number of elements\n";
            return;
        }
        $usels = 1 if (($dp > 1) && (@datf == 1) &&
$params{'tab_'.$table.'_load_datf'} !~ /\#/));

# expand all of the xxxf arrays to arrays of size $dp
        @ff = ();
        for ($i=0; $i < $dp; $i++)
        {
            $iplus = $i + 1;
            if (@ctlf > 1)
            {
                @cf[$i] = @ctlf[$i];
            }
            else
            {
                @cf[$i] = @ctlf[0];
                @cf[$i] =~ s/\#/$iplus/g;
            }
            if (@logf > 1)
            {
                @lf[$i] = @logf[$i];
            }
            else
            {
                @lf[$i] = @logf[0];
                @lf[$i] =~ s/\#/$iplus/g;
            }
            if (@badf > 1)
            {
                @bf[$i] = @badf[$i];
            }
            else
            {
                @bf[$i] = @badf[0];
                @bf[$i] =~ s/\#/$iplus/g;
            }
            if (@datf > 1)
            {
                @daf[$i] = @datf[$i];
            }
            else
            {
               # if load_degpar is larger than the
number partitions
               # and we are using pipes,
               # then load_degpar has to be a multiple
of the number
               # of partition and
load_degpar/#partitions sets of
               # pipes will be setup.
               # The goal is to maintain only one
pipe/flatfile per
               # sqlldr.

               if (($params{'tab_'.$table.'_#part'} >
0) &&
                   ($dp >
$params{'tab_'.$table.'_#part'})) {
                   if ($dp %
$params{'tab_'.$table.'_#part'} == 0) {
                       if (@datf[0] =~ /\#.*\#/) {
                           # set number

$snum=int($i/$params{'tab_'.$table.'_#part'}) + 1;
                           # partiton number

$pnum=$i%$params{'tab_'.$table.'_#part'} + 1;
                           @daf[$i] = @datf[0];
                           @daf[$i] =~ s/\#/$snum/;
                           @daf[$i] =~ s/\#/$pnum/;
                       } else {
                           @daf[$i] = @datf[0];
                           @daf[$i] =~ s/\#/$iplus/g;
                       }
                   }
                   else
                   {
                       print "Error: #partitons for
table $table does not divide load_degpar for the
table.\n";
                       print "degree is $dp, number of
partitions is $params{'tab_'.$table.'_#part'}.\n";
                       return;
                   }
               }
               else
               {
                   # assign
                   @daf[$i] = @datf[0];
                   @daf[$i] =~ s/\#/$iplus/g;
               }
            }
            if (@disf > 1)
            {
                @dif[$i] = @disf[$i];
            }
            else
            {
                @dif[$i] = @disf[0];
                @dif[$i] =~ s/\#/$iplus/g;
            }
            if (@filf > 1)
            {
                @ff[$i] = @filf[$i];
            }
            else
            {
                # round robining - temp fix
                @ff[$i] = @filf[0] if (@filf);
                if (defined
$params{$params{'tab_'.$table.'_ts'}.'_#files'}) {
                    $numfil =
$params{$params{'tab_'.$table.'_ts'}.'_#files'};
                    $filnum = $i % $numfil + 1;
                    @ff[$i] =~ s/\#/$filnum/g;
```

```
                } else {
                   @ff[$i] =~ s/\#/$iplus/g;
                }
            }
        }

        $control = sprintf ("%s%s",
$params{'load_controlfile_area'}, @cf[0]);
        $control =~ s/\?/$ENV{'ORACLE_HOME'}/g;

        $ff_path =
$params{'load_dbgen_'.$table.'_output_prefix'};
        $numchild =
$params{'load_dbgen_'.$table.'_option_C'};
        $pname = $params{'tab_'.$table.'_partnames'};
        @pname = split(/,/, $pname);
# create the controlfiles (fixed-length fields or
delimited records)
#       if ($params{'skip_mk_ldctlfs'} !~ /$table/)
#        {
          $npart = (defined
$params{'tab_'.$table.'_#part'}) ?
$params{'tab_'.$table.'_#part'} : 1;
          for ($ictl=1; $ictl<=$npart; $ictl++) {
            open (CTLFILE, ">$control");
            print CTLFILE "---\n";
            print CTLFILE "---  $table.ctl for
delimited records\n" if ($ud == 1);
            print CTLFILE "---  $table.ctl for fixed-
length fields\n" if ($ud == 0);
            print CTLFILE "---\n\n";
            if (!($pre72) && defined
$params{'tab_'.$table.'_loadextent'})
            {
              print CTLFILE "options\n";
              print CTLFILE "(\n";
              print CTLFILE "storage = (initial
$params{'tab_'.$table.'_loadextent'} next
$params{'tab_'.$table.'_loadextent'})\n";
              print CTLFILE ")\n";
            }
            print CTLFILE "unrecoverable\n" if
($params{'load_unrecoverable'} =~ /[tT][rR][uU][eE]/);
            print CTLFILE "load\n";
            print CTLFILE "-- This is where INFILE
should go.\n";
            if (($up)) && ($dbgpar)) {
              for
($jpart=1;$jpart<=$numchild;$jpart++) {
                  print CTLFILE "INFILE
'".$ff_path.$jpart."."$ictl."'\n";
              }
            }
            print CTLFILE
"$params{'load_insert_type'}\n";
            print CTLFILE "into table $table\n";
            if ($dgp) {
              print CTLFILE "partition ($pname[$ictl-
1])\n";
            }
            print CTLFILE
"$params{'load_insert_type'}\n";
            print CTLFILE "fields terminated by
$params{'load_field_terminator'}\n" if ($ud == 1);
            print CTLFILE "(\n";
            @tab_collist = split(/,/,
$params{'tab_'.$table.'_columns'});
            while ($col = shift(@tab_collist))
            {
              (@tab_collist == 0) ? ($addcomma = "")
: ($addcomma = ",");
              $pos = "";
              $pos = sprintf ("position
(%s)",$params{'tab_'.$table.'_'.$col.'_pos'}) if ($ud
== 0);
              $ctlline = sprintf ("  %-20s %s
%s$addcomma", $col, $pos,
$params{'tab_'.$table.'_'.$col.'_loadcolx'});
              print CTLFILE "$ctlline\n";
            }
            print CTLFILE ")\n";
            close (CTLFILE);
            if ($npart > 1) {
              $newi=$ictl+1;
              $control =~ s/$table$ictl/$table$newi/;
            }
          }
#       }

# create all of the "executable" load sections

# These variables will hold for all loaders for a
given table
        $upwd = "";
        $parbool = "";
        $dirbool = "";
        $silent = "";
        $dismax = "";
        $file = "";
        $errors = "";
        $rows = "";
        $bsize = "";
        $load = "";
        $skip = "";
        $dbgopt = "";
        $dbgtab = "";

        if ($up) {
            if ($table =~ /lineitem/)
            {
               $dbgtab = "L";
            }
            elsif ($table =~ /orders/)
            {
               $dbgtab = "O";
            }
            elsif ($table =~ /partsupp/)
            {
               $dbgtab = "S";
            }
            elsif ($table =~ /part/)
            {
               $dbgtab = "P";
            }
            elsif ($table =~ /customer/)
            {
               $dbgtab = "c";
            }
            elsif ($table =~ /supplier/)
            {
               $dbgtab = "s";
            }
            elsif ($table =~ /nation/)
            {
               $dbgtab = "n";
            }
            elsif ($table =~ /region/)
            {
               $dbgtab = "r";
            }
            $dbgopt .= "-T ".$dbgtab." ";
            $dbgopt .= "-s ". $params{'scale_factor'}
. " ";
            $dbgopt .= "-C ";
            $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?
                        $params{'load_dbgen_'.$table.'_
option_C'} :
                        $params{'load_dbgen_def_option_
C'});
            if ($dgp) {
               $dbgopt .= " -p -i ";
               $dbgopt .= (defined
$params{'load_dbgen_'.$table.'_input_params'}) ?
$params{'load_dbgen_'.$table.'_input_params'} :
$params{'load_dbgen_def_input_params'};
            }
        }

        $upwd = "$params{'user'}/$params{'passwd'}";
        $parbool = " parallel=true" if
(($params{'tab_'.$table.'_load_parallel'} =~ /true/)
|| ($dp > 1));
        $dirbool = " direct=true" if
(($params{'tab_'.$table.'_load_direct'} =~
/[Tt][Rr][Uu][Ee]/) || $parbool);
        $silent = "
silent=$params{'tab_'.$table.'_load_silent'}" if
defined $params{'tab_'.$table.'_load_silent'};
        $dismax = "
discardmax=$params{'tab_'.$table.'_load_dismax'}" if
defined $params{'tab_'.$table.'_load_dismax'};
        $errors = "
errors=$params{'tab_'.$table.'_load_errors'}" if
defined $params{'tab_'.$table.'_load_errors'};
        $rows = "
rows=$params{'tab_'.$table.'_load_rows'}" if defined
$params{'tab_'.$table.'_load_rows'};
        $bsize = "
bindsize=$params{'tab_'.$table.'_load_bsize'}" if
defined $params{'tab_'.$table.'_load_bsize'};
```

```
        if ($usels)
        {
            $split =
int($params{'tab_'.$table.'_#rows'} / $dp);
            $extra =
int($params{'tab_'.$table.'_#rows'} % $dp);
            $loadval = $split;
            $skipval = 0;
        }

        # if using pipes, create the pipes

        if($up)
        {
            for ($i=0;$i<$dp;$i++) {
                $cmd = sprintf("rm -f %s%s\nmknod %s%s
p\n",$params{'load_flatfile_area'},@daf[$i],$params{'l
oad_flatfile_area'},@daf[$i]);
#               $cmd = sprintf("%s%s
p\n",$params{'load_flatfile_area'},@daf[$i]);
                &dump1("*sh");
            }
            &dump0("*wait");

            # start the dbgens

            if ($dgp) {
                $nset = int($dp /
$params{'tab_'.$table.'_#part'});
                $nperset = int($dbgpar / $nset);
            }

            for ($i=0;$i<$dbgpar;$i++) {
                $cmd = $params{'load_flatfile_area'} .
"\n";
                $cmd2 = sprintf("-f %s -S %d", $dbgopt,
($i + 1));
                if ($dgp) {
                    # if the output prefix name is
specified with #
                    # divide output file names into
$dp/#numpart sets
                    $fpre = (defined
$params{'load_dbgen_'. $table.'_output_prefix'}) ?
$params{'load_dbgen_'.$table.'_output_prefix'} :
$params{'load_dbgen_def_output_prefix'};
                    if ($fpre =~ /\#/) {
                        $snum = int($i/$nperset) + 1;
                        $fpre =~ s/\#/$snum/g;
                    }
                    $cmd2 .= sprintf(" -o %s ", $fpre);
                }
                $cmd2 .= "\n";
                &dump2("*dbgen");
            }
        }

        $control = "";
        $log = "";
        $bad = "";
        $data = "";
        $discard = "";
        $file = "";
        $load = "";
        $loadval = "";
        $skip = "";

        for ($i=0; $i < $dp; $i++)
        {
            &advmulti();

            $control = sprintf (" control=%s%s",
$params{'load_controlfile_area'}, @cf[$i]);
            $control = sprintf (" control=/host%s%s",
$params{'load_controlfile_area'}, @cf[$i]) if
($params{'special_machine'} eq 'ncube');
            $log = sprintf (" log=%s%s",
$params{'load_otherfile_area'}, @lf[$i]) if $lfexist;
            $bad = sprintf (" bad=%s%s",
$params{'load_otherfile_area'}, @bf[$i]) if $bfexist;
            $data = sprintf (" data=%s%s",
$params{'load_flatfile_area'}, @daf[$i]) if $dafexist;
            $discard = sprintf (" discard=%s%s",
$params{'load_otherfile_area'}, @dif[$i]) if
$difexist;
            $file = sprintf (" file=%s%s",
$params{$curts.'_area'}, @ff[$i]) if $ffexist;

            if ($usels)
            {
                $loadval = $loadval + 1 if ($extra >
$i);
                $load = sprintf (" load=%d",
$loadval);
                $skip = sprintf (" skip=%d",
$skipval);
                $skipval = $skipval + $loadval;
                $loadval = $split;
            }
            if ($up)
            {
                $load = " load=99999999";
            }

            $cmd = sprintf
("%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n",$upwd,$control,$
log,$bad,$data,$discard,$dismax,$skip,$load,$errors,$r
ows,$bsize,$silent,$dirbool,$parbool,$file);
            &dump1("*load");
            &dump0("*wait") if ((!$parbool) &&
(!defined $params{'load_no_waits'}));
        }
        &dump0("*wait") if !defined
$params{'load_no_waits'};

        if($up)
        {
            for ($i=0;$i<$dp;$i++) {
                $cmd = sprintf("rm -f
%s%s\n",$params{'load_flatfile_area'},@daf[$i]);
                &dump1("*sh");
            }
            &dump0("*wait");
        }
    }
}


sub plcre
{

&dump0("############################################
##############################");
    &dump0("# NT Raw Partition and Setlink Creation
Phase");
    &time0("Begin NT Raw Partition and Setlink
creation");

# create NT Raw Partition and Setlink input files
    $lnksfile = $params{'plcre_setlinks_input_file'};
    open (LNKSFILE, ">$lnksfile");
    $drivenum = $params{'plcre_drivenums'};
    $md = $drivenum =~ tr/,/,/;
    @drivenum = split(/,/,$params{'plcre_drivenums'});
if ($params{'plcre_recreate_extended_partitions' =~
/[tT][rR][uU][eE]/})
    {
        foreach $drivenum (@drivenum)
        {
            &recreate_drive_extended_part;
        }
        if (defined $params{'plcre_log_drivenum'})
        {
            &recreate_drive_extended_part;
        }
    }
    $io_control_files = $params{'io_control_files'};
    $io_control_files =~ s/[\(\"\\\.\)]*//g;
    @io_control_files = split(/,/,$io_control_files);
    $d = 0;
    foreach $file (@io_control_files)
    {
        $size = $params{'plcre_control_file_size'};
        $size =~ s/[Mm]*//g;
        &create_drive_part_file;
    }
    foreach $ts_entry (@ts_all)
    {
        &create_drive_part("$ts_entry") if ($ts_entry
=~ /ts_sys/ || $ts_entry =~ /ts_log/);
    }
    $d = 0;
    foreach $ts_entry (@ts_most)
    {
        &create_drive_part("$ts_entry");
    }
    close (LNKSFILE);
    &dump0("*wait");
    &time0("Done NT Raw Partition creation");
```

```
    $cmd = "setlinks /F:" . $lnksfile . "\n";
    &dump1("*sh");
    &dump0("*wait");
    &time0("Done Setlink links creation");
    &time0("Done NT Raw Partition and Setlink
creation");
} # end of plcre



sub sccre
{

&dump0("##############################################
##############################");
    &dump0("# Schema Creation Phase");
    &time0("Begin schema creation");

# Create user
    if ($params{'skip_create_user'} !~ /true/)
    {
        &dump0("# creating $params{'user'} user");
        $cmd .= "drop user $params{'user'}
cascade;\n";
        $cmd .= "grant $params{'privileges'}\n";
        $cmd .= "  to $params{'user'} identified by
$params{'passwd'};\n";
        $cmd .= "@?/rdbms/admin/utlxplan;\n" if (($os
cmp "unix") == 0);
        &dump1("*sql");
    }
    &dump0("*wait");

# Create data tablespaces (including datafiles and
tables)
    &dump0("# creating data tablespaces, datafiles,
and tables");

# create tablespaces with initial datafile
    foreach $ts_entry (@ts_data)
    {
        &create_ts("$ts_entry");
    }
    foreach $ts_entry (@ts_index)
    {
        &create_ts("$ts_entry");
    }
    foreach $ts_entry (@ts_temp)
    {
        $ts_temporary = "temporary" if !$pre73;
        &create_ts("$ts_entry");
    }
    &dump0("*wait");

# create remaining datafiles for the tablespaces
    foreach $ts_entry (@ts_data)
    {
        &add_dfs("$ts_entry");
    }
    foreach $ts_entry (@ts_index)
    {
        &add_dfs("$ts_entry");
    }
    if ($params{'skip_ts'} !~ /temp/)
    {
        foreach $ts_entry (@ts_temp)
        {
            &add_dfs("$ts_entry");
        }
    }
    &dump0("*wait");

    if ($params{'skip_create_tables'} !~ /true/)
    {
        &create_clusters();
        &dump0("*wait");
        &create_objects('tab',@tab_list) if
defined(@tab_list);
        &dump0("*wait");
    }

    if ($params{'skip_ts'} !~ /temp/)
    {
# Alter user's temporary tablespace
        &dump0("# altering $params{'user'}'s temporary
tablespace");
        $ts_entry = shift (@ts_temp);
        $cmd = "alter user $params{'user'} temporary
tablespace $ts_entry;\n";
        &dump1("*sql");
        &dump0("*wait");
    }
    if ($params{'skip_ts'} !~ /data/)
    {
# Alter user's default tablespace
        &dump0("# altering $params{'user'}'s default
tablespace");
        $ts_entry = shift (@ts_data);
        $cmd = "alter user $params{'user'} default
tablespace $ts_entry;\n";
        &dump1("*sql");
        &dump0("*wait");
    }
    &time0("Done schema creation");
}


sub sctso
{

    if ($phaselist !~ /sccre/)
    {
        &dump0("#####################################
##################################");
        &dump0("# Schema Creation Phase – datafiles
only (no tables or users)");

# Create data tablespaces (including datafiles)
        &dump0("# creating data tablespaces,
datafiles");

# create tablespaces with initial datafile
        foreach $ts_entry (@ts_data)
        {
            &create_ts("$ts_entry");
        }
        foreach $ts_entry (@ts_index)
        {
            &create_ts("$ts_entry");
        }
        foreach $ts_entry (@ts_temp)
        {
            $ts_temporary = "temporary" if !$pre73;
            &create_ts("$ts_entry");
        }
        &dump0("*wait");

# create remaining datafiles for the tablespaces
        foreach $ts_entry (@ts_data)
        {
            &add_dfs("$ts_entry");
        }
        foreach $ts_entry (@ts_index)
        {
            &add_dfs("$ts_entry");
        }
        if ($params{'skip_ts'} !~ /temp/)
        {
            foreach $ts_entry (@ts_temp)
            {
                &add_dfs("$ts_entry");
            }
        }
        &dump0("*wait");

        if ($params{'skip_ts'} !~ /temp/)
        {
# Alter user's temporary tablespace
#           &dump0("# altering $params{'user'}'s
temporary tablespace");
#           $ts_entry = shift (@ts_temp);
#           $cmd = "alter user $params{'user'}
temporary tablespace $ts_entry;\n";
#           &dump1("*sql");
#           &dump0("*wait");
        }
    }
}


sub scuto
{
    if ($phaselist !~ /sccre/)
    {

&dump0("##############################################
##############################");
        &dump0("# Schema Creation Phase – User and
```

```
Tables ONLY (no datafiles)");
        &time0("Begin creating user and tables (no
datafiles)");

        if ($params{'skip_create_user'} !~ /true/)
        {
            &dump0("# creating $params{'user'} user");
            $cmd .= "drop user $params{'user'}
cascade;\n";
            $cmd .= "grant $params{'privileges'}\n";
            $cmd .= "  to $params{'user'} identified
by $params{'passwd'};\n";
            &dump1("*sql");
        }
        &dump0("*wait");

        &create_clusters();
        &dump0("*wait");
        &create_objects('tab',@tab_list) if
defined(@tab_list);
        &dump0("*wait");

        if ($params{'skip_ts'} !~ /temp/)
        {
# Alter user's temporary tablespace
            &dump0("# altering $params{'user'}'s temp
tablespace");
            $ts_entry = shift (@ts_temp);
            $cmd = "alter user $params{'user'}
temporary tablespace $ts_entry;\n";
            &dump1("*sql");
            &dump0("*wait");
        }
# Alter user's default tablespace
        if ($params{'skip_ts'} !~ /data/)
        {
            &dump0("# altering $params{'user'}'s
default tablespace");
            $ts_entry = shift (@ts_data);
            $cmd = "alter user $params{'user'} default
tablespace $ts_entry;\n";
            &dump1("*sql");
            &dump0("*wait");
        }
        $cmd .= "connect tpcd/tpcd\n";
        $cmd .= '@' . "$params{'dd_sql_area'}" .
"utlxplan.sql;\n";
        &dump1("*sql");
        &dump0("*wait");
        &time0("Done creating user and tables (no
datafiles)");
    }
}


sub scuvo
{
    if ($phaselist =~ /scuvo/)
    {

&dump0("#############################################
##############################");
        &dump0("# Schema Creation Phase - Views ONLY
(no datafiles)");

        if (defined(@tablelog_list))
        {

&dump0("#############################################
##############################");
            &dump0("# First I will create the
materialized log for the base tables");
            &create_objects('viewlog',@tablelog_list);
            &dump0("*wait");
        }

&dump0("#############################################
##############################");
        &dump0("# Now I can create the views with the
corresponding materialized logs");

        $cmd .= "connect
$params{'user'}/$params{'passwd'};\n";

        foreach $view (@view_list)
        {
            @viewlogs =
split(/,/,$params{'view_'.$view.'_viewlogs'});
            foreach $viewlog (@viewlogs)
            {
                $ob_type = 'viewlog';
                $object = $viewlog;
                if (!(defined
$params{'created_'.$object}) )
                {
                    $params{'created_'.$object} =
$object;
                    &create_object;
                }
            }
            $ob_type = 'view';
            $object = $view;
            &create_object;
        }
        &dump1("*sql");



#       &create_objects('view',@view_list) if
defined(@view_list);
#       &create_objects('viewlog',@viewlog_list) if
defined(@tablelog_list);

        &dump0("*wait")
        }
}

sub dsffs
{

&dump0("#############################################
##############################");
    &dump0("# Flatfile Distribution Phase");
    &time0("Begin data population");

    # &recycle_db() if ($params{'startup_db_dapop'});

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);

}
sub dapop # Meikel's version
{

&dump0("#############################################
##############################");
    &dump0("# Database Population Phase");
    &time0("Begin data population");

    # &recycle_db() if ($params{'startup_db_dapop'});


    # find out how many nodes we have
    if (!(defined(@ops_nodes))) {
        $nnodes=1
    } else {
        $nnodes= ($#ops_nodes + 1);
    }

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        &time0("Begin $table load");
        $curts = $params{'tab_'.$table.'_ts'};

# see if we are using pipes

        $up = $params{'load_use_pipes'} =~ /$table/ ?
1 : 0;

        $dp = $params{'tab_'.$table.'_load_degpar'};
        $dgp = ($params{'load_dbgen_partition'} =~
/$table/) ? 1 : 0;
        $dbgpar =
$params{'load_dbgen_'.$table.'_option_C'};

# These variables will hold for all loaders for a
given table
        $upwd = "";
        $parbool = "";
```

```
        $dirbool = "";
        $silent = "";
        $dismax = "";
        $errors = "";
        $rows = "";
        $bsize = "";
        $load = "";
        $skip = "";
       $dbgopt = "";
       $dbgtab = "";

        $upwd = "$params{'user'}/$params{'passwd'}";
        $parbool = " parallel=true" if
(($params{'tab_'.$table.'_load_parallel'} =~ /true/)
|| ($dp > 1));
        $dirbool = " direct=true" if
(($params{'tab_'.$table.'_load_direct'} =~
/[Tt][Rr][Uu][Ee]/) || $parbool);
        $silent = "
silent=$params{'tab_'.$table.'_load_silent'}" if
defined $params{'tab_'.$table.'_load_silent'};
        $dismax = "
discardmax=$params{'tab_'.$table.'_load_dismax'}" if
defined $params{'tab_'.$table.'_load_dismax'};
        $errors = "
errors=$params{'tab_'.$table.'_load_errors'}" if
defined $params{'tab_'.$table.'_load_errors'};
        $rows = "
rows=$params{'tab_'.$table.'_load_rows'}" if defined
$params{'tab_'.$table.'_load_rows'};
        $bsize = "
bindsize=$params{'tab_'.$table.'_load_bsize'}" if
defined $params{'tab_'.$table.'_load_bsize'};

        $ff_path =
$params{'load_flatfile_area'}.$params{'load_dbgen_'.$t
able.'_output_prefix'};
        $pname = $params{'tab_'.$table.'_partnames'};
        $pname =~ s/\#//g;
        print $pname."--".@pname;
        @pname = split(/\,/,$pname);
        $npart = ((defined
$params{'tab_'.$table.'_#part'}) && $dgp) ?
            $params{'tab_'.$table.'_#part'} : 1;
        $ctlarea = $params{'load_controlfile_area'};
        $logarea = $params{'load_otherfile_area'};

        if ($dbgpar*$npart <= $dp)
        {
            $dpmax = $dbgpar*$npart;
            $numchild = 1;
            $remchild = 0;
        }
        else
        {
            $dpmax = $dp;
            $numchild = int(($dbgpar*$npart)/$dp); #
numchild is the number of infiles of one loader
            $remchild = ($dbgpar*$npart) - ($numchild
* $dp);
            #print "dp numchild remchild ".$dp."
".$numchild." ".$remchild."\n";
            $loaders_p_part=$dp/$npart;
            #print "dp npart loaders per partition
".$loaders_p_part." ".$dp." ".$npart."\n";
            $ninfiles=int($dbgpar/$loaders_p_part);

$ninfiles_last=$ninfiles+$dbgpar-($ninfiles *
$loaders_p_part);
            #print "infiles lastloader ".$ninfiles."
".$ninfiles_last."\n";
        }

        # if we load on an ops we want to load each
partition only on one node

        $npartitions_per_node = int($npart / $nnodes);
# #partition/#nodes
        $rempartitions = $npart -
($npartitions_per_node * $nnodes); #reminder of the
partitions
        print "nnodes $nnodes npart $npart table
$table npartitions_per_node $npartitions_per_node
rempartitions $rempartitions\n";

        $node = 0;
         $idbgpar = 0;
        $ipart=1;
        $iipart=1;
        if ($dp > $npart) {
            print "$dp $npart\n";
            $sqlldr_per_partition = int ($dp /
$npart); # load_degpar / #partitions
            print "** dp $dp npart $npart
sqlldr_per_partition $sqlldr_per_partition\n";
            $infiles_per_sqlldr = int ($dbgpar /
$sqlldr_per_partition ); #_C / sqlldr_per_partition
            $sqlldr_per_partition_rest = $dbgpar -
$sqlldr_per_partition * $infiles_per_sqlldr;
            print "** dbgpar $dbgpar
sqlldr_per_partition $sqlldr_per_partition\n";
            print "** sqlldr_per_partition
$sqlldr_per_partition sqlldr_per_partition_rest
$sqlldr_per_partition_rest infiles_per_sqlldr
$infiles_per_sqlldr \n";
        }
        else{
             $sqlldr_per_partition = 1;
             $infiles_per_sqlldr = int ($dbgpar /
$sqlldr_per_partition ); #_C / sqlldr_per_partition
         }

        $partcount = 1;
        if (defined(@ops_nodes) && ($dpmax > 1)){
            &sqlldr_ctl_ops; # this is a special
controlfile generation for OPS
        }
        else {
        for ($idp=1; $idp<=$dpmax; $idp++)
        {
            if ($partcount > $sqlldr_per_partition) {
                $partcount = 1;
            }
            if (defined(@ops_nodes)){
                $node = $node + 1;
                if ($node > $#ops_nodes+1) {
                    $node = 1;
                }
            }
            else {
                $node = "";
            }
            #$idbgpar = 0;
# create all of the "executable" load sections
            #$ipart= int((($idp-
1)*$numchild)/$dbgpar)+1; # this is the first
partition of the new series
            &advmulti();
            $ctlfile =
$params{'tab_'.$table.'_load_ctlf'};
            $ctlfile =~ s/\#/$idp/g;
            $control = $ctlarea.$ctlfile;
            if ($dgp) # load this table in partition
mode
            {
                #$control = sprintf (" %s%s_%d_%d.ctl",
$ctlarea, $table, $idp, $ipart);
                $log = sprintf (" %s%s_%d_%d.log",
$logarea, $table, $idp, $ipart);
            }
            else
            {
                if ($dbgpar == 1) #dbgpar is option _C
for this table
                {
                    #$control = sprintf (" %s%s.ctl",
$ctlarea, $table);
                    $log = sprintf (" %s%s.log",
$logarea, $table);
                }
                else
                {
                    #$control = sprintf ("
%s%s_%d.ctl", $ctlarea, $table, $idp);
                    $log = sprintf (" %s%s_%d.log",
$logarea, $table, $idp);
                }
            }
            $cmd = sprintf ("%s control=%s%s%s
log=%s%s%s%s%s%s%s%s\n",

$upwd,$control,$skip,$load,$log,$errors,$rows,

$bsize,$silent,$dirbool,$parbool);
            &dump1("*load$node");
            &dump0("*wait") if ((!$parbool) &&
(!defined $params{'load_no_waits'}));

# create the controlfiles (fixed-length fields or
delimited records)
```

```
            if ($params{'skip_mk_ldctlfs'} !~
/$table/)
            {
                &load_ctl_head;
                $infiles = 0;
                $badfiles = 0;
                $discardfiles = 0;
                $intopartitions = "BEGINNING"; # this
will collect the partition numbers that will be
served.
                if ($dp < $npart) {
                    if ($idp <= $remchild) {
                        $nchild = $numchild+1;
                    }else{
                        $nchild = $numchild;
                    }
                }else{
                    if ($partcount <=
$sqlldr_per_partition_rest) {
                        $nchild =
$infiles_per_sqlldr+1;
                    }else{
                        $nchild = $infiles_per_sqlldr;
                    }
                }
                for ($jpart=1; $jpart<=$nchild;
$jpart++)
                {
                    print "jpart $jpart\n";
                    $idbgpar = 0 if $idbgpar >=
$dbgpar;
                    #$ipart = int((($idp-
1)*$nchild+$jpart)/$dbgpar);
                    #$ipart = $ipart+1 if ((($idp-
1)*$nchild+$jpart)-$ipart*$dbgpar)>0);
                    if ($dgp)
                    {
                        $pn=@ipart[$node];
                        $partitionname = $pname.$pn;
                        if (!($intopartitions =~
/$partitionname/)) {
                            if ($intopartitions =~
/BEGINNING/)
                            {
                                $intopartitions =
$partitionname;
                            }
                            else {
                                $intopartitions .=
",".$partitionname;}
                        }
                        @infile[$infiles++] = "'" .
$ff_path . "_" . ++$idbgpar .
                                "." . $ipart . "'";
                        @badfile[$badfiles++] = "'" .
$ff_path . "_" . $idbgpar .
                                "_" . $ipart . ".bad'";
                        @discardfile[$discardfiles++] =
"'" . $ff_path . "_" . $idbgpar .
                                "_" . $ipart . ".dsc'";
                        #if ($remchild-- > 0)
                        #{
                        #    @infile[$infiles++] = "'"
. $ff_path . "_" . ++$idbgpar .
                        #            "." . $ipart . "'";
                        #    @badfile[$badfiles++] =
"'" . $ff_path . "_" . $idbgpar .
                        #            "_" . $ipart . ".bad'";
                        #
@discardfile[$discardfiles++] = "'" . $ff_path . "_" .
$idbgpar .
                        #            "_" . $ipart . ".dsc'";
                        #}
                    }
                    elsif ($dbgpar == 1)
                    {
                        @infile[$infiles++] = "'" .
$ff_path . ".tbl'";
                        @badfile[$badfiles++] = "'" .
$ff_path . ".bad'";
                        @discardfile[$discardfiles++] =
"'" . $ff_path . ".dsc'";
                    }
                    else
                    {
                        @infile[$infiles++] = "'" .
$ff_path . ".tbl." . ++$idbgpar . "'";
                        @badfile[$badfiles++] = "'" .
$ff_path . "_" . $idbgpar . ".bad'";
                        @discardfile[$discardfiles++] =
"'" . $ff_path . "_" . $idbgpar .
                                ".dsc'";
                        if ($remchild-- > 0)
                        {
                            @infile[$infiles++] = "'" .
$ff_path . ".tbl." . ++$idbgpar . "'";
                            @badfile[$badfiles++] = "'"
. $ff_path . "_" . $idbgpar . ".bad'";

@discardfile[$discardfiles++] = "'" . $ff_path . "_" .
$idbgpar .
                                ".dsc'";
                        }
                    }
                    #print "vorher jpart $jpart ipart
$ipart iipart $iipart dbgpar $dbgpar\n";
                    if ($iipart >= $dbgpar){
                        $iipart=0;
                        $ipart=$ipart+1;
                    }
                    $iipart=$iipart+1;
                    #print "nacher jpart $jpart ipart
$ipart iipart $iipart dbgpar $dbgpar\n";
                }
                --$infiles;
                if ($up)
                {
                    print CTLFILE "INFILE ";
                    for ($fpart=0; $fpart<=$infiles;
$fpart++)
                    {
                        print CTLFILE @infile[$fpart] .
", " if $fpart < $infiles;
                        print CTLFILE @infile[$fpart] .
" " if $fpart == $infiles;
                    }
                    print CTLFILE "BADFILE " .
@badfile[0] . " ";
                    print CTLFILE "DISCARDFILE " .
@discardfile[0] . "\n";
                }
                else
                {
                    for ($fpart=0; $fpart<=$infiles;
$fpart++)
                    {
                        print CTLFILE "INFILE " .
@infile[$fpart] . " ";
                        print CTLFILE "BADFILE " .
@badfile[$fpart] . " ";
                        print CTLFILE "DISCARDFILE " .
@discardfile[$fpart] . "\n";
                    }
                }
                print CTLFILE
"$params{'load_insert_type'}\n";
                print CTLFILE "into table $table\n";
                if ($dgp)
                {
                    #print CTLFILE "partition
($pname$ipart)\n" if
!($params{'tab_'.$table.'_parttype'} =~ /hash/);
                    #print CTLFILE "partition
($intopartitions)\n" if
!($params{'tab_'.$table.'_parttype'} =~ /hash/);
                }
                &load_ctl_tail;
            }
            $partcount = $partcount + 1;
        }
        &dump0("*wait") if !defined
$params{'load_no_waits'} && $dgp;
        &time0("End of load for Partition: $ipart for
Table: $table") if ($dgp);

        &dump0("*wait") if !defined
$params{'load_no_waits'};
    }   &time0("End $table load");
    }
    &dump0("*wait");
    &time0("Done data population");
} # end of dapop


sub sqlldr_ctl_ops
{
    $controlfiles_per_node = $dpmax / $nnodes;
    #print "controlfiles_per_node
$controlfiles_per_node nnodes $nnodes\n";
    for ($i=1; $i<=$nnodes; $i++){
```

```
        @ipart[$i] = (($i-1)*$npartitions_per_node)+1;
        @iipart[$i] = 1;
        #print "Offset for node $i is: @ipart[$i]\n";
        $file_offset[i] = 1;
    }

    for ($idpp=1; $idpp<=$controlfiles_per_node;
$idpp++) {
    for ($node=1; $node<=$nnodes; $node++)
    {
        #print "idpp $idpp node $node
infiles_per_sqlldr $infiles_per_sqlldr\n";
        $idp = (($node-1)*$controlfiles_per_node) +
$idpp;
        if ($partcount > $sqlldr_per_partition) {
            $partcount = 1;
        }
        #if (defined(@ops_nodes)){
        #    $node = $node + 1;
        #    if ($node > $#ops_nodes+1) {
        #       $node = 1;
        #    }
        #}
        #else {
        #    $node = "";
        #}
        #$idbgpar = 0;
# create all of the "executable" load sections
        #@ipart[$node]= int((($idp-
1)*$numchild)/$dbgpar)+1; # this is the first
partition of the new series
        &advmulti();
        $ctlfile =
$params{'tab_'.$table.'_load_ctlf'};
        $ctlfile =~ s/\#/$idp/g;
        $control = $ctlarea.$ctlfile;
        if ($dgp) # load this table in partition mode
        {
            #$control = sprintf (" %s%s_%d_%d.ctl",
$ctlarea, $table, $idp, @ipart[$node]);
            $log = sprintf (" %s%s_%d_%d.log",
$logarea, $table, $idp, @ipart[$node]);
        }
        else
        {
            if ($dbgpar == 1) #dbgpar is option _C for
this table
            {
                #$control = sprintf (" %s%s.ctl",
$ctlarea, $table);
                $log = sprintf (" %s%s.log", $logarea,
$table);
            }
            else
            {
                #$control = sprintf (" %s%s_%d.ctl",
$ctlarea, $table, $idp);
                $log = sprintf (" %s%s_%d.log",
$logarea, $table, $idp);
            }
        }
        $cmd = sprintf ("%s control=%s%s%s
log=%s%s%s%s%s%s%s\n",
                        $upwd,$control,$skip,$load,$log
,$errors,$rows,
                        $bsize,$silent,$dirbool,$parboo
l);
        &dump1("*load$node");
        &dump0("*wait") if ((!$parbool) && (!defined
$params{'load_no_waits'}));

# create the controlfiles (fixed-length fields or
delimited records)
        if ($params{'skip_mk_ldctlfs'} !~ /$table/)
        {
            &load_ctl_head;
            $infiles = 0;
            $badfiles = 0;
            $discardfiles = 0;
            $intopartitions = "BEGINNING"; # this will
collect the partition numbers that will be served.
            if ($dp < $npart) {
                if ($idp <= $remchild) {
                    $nchild = $numchild+1;
                }else{
                    $nchild = $numchild;
                }
            }else{
                if ($partcount <=
$sqlldr_per_partition_rest) {
                    $nchild = $infiles_per_sqlldr+1;
                }else{
                    $nchild = $infiles_per_sqlldr;
                }
            }
            for ($jpart=1; $jpart<=$nchild; $jpart++)
            {
                $file_offset[$node]++;
                if ( $file_offset[$node] > $dbgpar) {
                    $file_offset[$node] = 1;
                }
                #print "Offset
$node:".$file_offset[$node]."\n";
                #print "ipart @ipart[$node]\n";
                $idbgpar = 0 if $idbgpar >= $dbgpar;
                #@ipart[$node] = int((($idp-
1)*$nchild+$jpart)/$dbgpar);
                #@ipart[$node] = @ipart[$node]+1 if
((($idp-1)*$nchild+$jpart)-@ipart[$node]*$dbgpar)>0);
                if ($dgp) # load in partition mode
                {
                    #print @ipart[$node]."\n";
                    $pn=@ipart[$node];
                    $partitionname = $pname.$pn;
                    if (!($intopartitions =~
/$partitionname/)) {
                        if ($intopartitions =~
/BEGINNING/)
                        {
                            $intopartitions =
$partitionname;
                        }
                        else {
                            $intopartitions .=
",".$partitionname;}
                        }
                    #print "before infile: node $node
file_offset $file_offset[$node]\n";
                    @infile[$infiles++] = "'" .
$ff_path . "_" . $file_offset[$node] .
                        "." . @ipart[$node] . "'";
                    @badfile[$badfiles++] = "'" .
$ff_path . "_" . $file_offset[$node] .
                        "_" . @ipart[$node] . ".bad'";
                    @discardfile[$discardfiles++] = "'"
. $ff_path . "_" . $file_offset[$node] .
                        "_" . @ipart[$node] . ".dsc'";
                    #if ($remchild-- > 0)
                    #{
                    #    @infile[$infiles++] = "'" .
$ff_path . "_" . ++$idbgpar .
                    #       "." . @ipart[$node] .
"'";
                    #    @badfile[$badfiles++] = "'" .
$ff_path . "_" . $idbgpar .
                    #    "_" . @ipart[$node] . ".bad'";
                    #    @discardfile[$discardfiles++]
= "'" . $ff_path . "_" . $idbgpar .
                    #   "_" . @ipart[$node] . ".dsc'";
                    #}
                }
                elsif ($dbgpar == 1)
                {
                    @infile[$infiles++] = "'" .
$ff_path . ".tbl'";
                    @badfile[$badfiles++] = "'" .
$ff_path . ".bad'";
                    @discardfile[$discardfiles++] = "'"
. $ff_path . ".dsc'";
                }
                else
                {
                    @infile[$infiles++] = "'" .
$ff_path . ".tbl." . ++$idbgpar . "'";
                    @badfile[$badfiles++] = "'" .
$ff_path . "_" . $idbgpar . ".bad'";
                    @discardfile[$discardfiles++] = "'"
. $ff_path . "_" . $idbgpar .
                        ".dsc'";
                    if ($remchild-- > 0)
                    {
                        @infile[$infiles++] = "'" .
$ff_path . ".tbl." . ++$idbgpar . "'";
                        @badfile[$badfiles++] = "'" .
$ff_path . "_" . $idbgpar . ".bad'";
                        @discardfile[$discardfiles++] =
"'" . $ff_path . "_" . $idbgpar .
                            ".dsc'";
                    }
                }
```

```
                #print "vorher jpart $jpart ipart
@ipart[$node] iipart @iipart[$node] dbgpar $dbgpar\n";
                if (@iipart[$node] >= $dbgpar){
                    @iipart[$node]=0;
                    @ipart[$node]=@ipart[$node]+1;
                }
                @iipart[$node]=@iipart[$node]+1;
                #print "nacher jpart $jpart ipart
@ipart[$node] iipart @iipart[$node] dbgpar $dbgpar\n";
            }
            --$infiles;
            if ($up)
            {
                print CTLFILE "INFILE ";
                for ($fpart=0; $fpart<=$infiles;
$fpart++)
                {
                    print CTLFILE @infile[$fpart] . ",
" if $fpart < $infiles;
                    print CTLFILE @infile[$fpart] . " "
if $fpart == $infiles;
                }
                print CTLFILE "BADFILE " . @badfile[0]
. " ";
                print CTLFILE "DISCARDFILE " .
@discardfile[0] . "\n";
            }
            else
            {
                for ($fpart=0; $fpart<=$infiles;
$fpart++)
                {
                    print CTLFILE "INFILE " .
@infile[$fpart] . " ";
                    print CTLFILE "BADFILE " .
@badfile[$fpart] . " ";
                    print CTLFILE "DISCARDFILE " .
@discardfile[$fpart] . "\n";
                }
            }
            print CTLFILE
"$params{'load_insert_type'}\n";
            print CTLFILE "into table $table\n";
            if ($dgp)
            {
                #print CTLFILE "partition
($pname@ipart[$node])\n" if
!($params{'tab_'.$table.'_parttype'} =~ /hash/);
                print CTLFILE "partition
($intopartitions)\n" if
!($params{'tab_'.$table.'_parttype'} =~ /hash/);
            }
            &load_ctl_tail;
        }
        $partcount = $partcount + 1;
    }
}
    &dump0("*wait") if !defined
$params{'load_no_waits'} && $dgp;
    &time0("End of load for Partition: @ipart[$node]
for Table: $table") if ($dgp);

    &dump0("*wait") if !defined
$params{'load_no_waits'};
    &time0("End $table load");
}



sub dapop_frank ## copied from Frank's version
{

&dump0("############################################
##############################");
    &dump0("# Database Population Phase");
    &time0("Begin data population");

    $cur_inst = 1 if $multi;  # for possible locality
on SP2
    ($params{'load_type'} =~ /delim/) ? ($ud = 1) :
($ud = 0);
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    @load_tablelist = split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        &time0("Begin $table load");

        $curts = $params{'tab_'.$table.'_ts'};
        $dp = $params{'tab_'.$table.'_load_degpar'};
        $usels = 0;
        $lfexist = 0;
        $bfexist = 0;
        $dafexist = 0;
        $difexist = 0;
        $ffexist = 0;
        $up = 0;
        $dgp = 0;
        $dbgpar = 1;

        # see if we are using pipes

        $up = 1 if ($params{'load_use_pipes'} =~
/$table/);
        $dgp = 1 if ($params{'load_dbgen_partition'}
=~ /$table/);
        $dbgpar = $params{'load_dbgen_def_option_C'}
if defined $params{'load_dbgen_def_option_C'};
        $dbgpar =
$params{'load_dbgen_'.$table.'_option_C'} if defined
$params{'load_dbgen_'.$table.'_option_C'};

        $params{'tab_'.$table.'_load_ctlf'} =
$table.'.ctl' if !defined
$params{'tab_'.$table.'_load_ctlf'};
        @ctlf = split(/,/,
$params{'tab_'.$table.'_load_ctlf'});
        $err = "ctl" if ((@ctlf > 1) && (@ctlf !=
$dp));
        @logf = split(/,/,
$params{'tab_'.$table.'_load_logf'});
        $err = "log" if ((@logf > 1) && (@logf !=
$dp));
        $lfexist = 1 if @logf > 0;
        @badf = split(/,/,
$params{'tab_'.$table.'_load_badf'});
        $err = "bad" if ((@badf > 1) && (@badf !=
$dp));
        $bfexist = 1 if @badf > 0;
        @datf = split(/,/,
$params{'tab_'.$table.'_load_datf'});
        $err = "dat" if ((@datf > 1) && (@datf !=
$dp));
        $dafexist = 1 if @datf > 0;
        @disf = split(/,/,
$params{'tab_'.$table.'_load_disf'});
        $err = "dis" if ((@disf > 1) && (@disf !=
$dp));
        $difexist = 1 if @diff > 0;
       @filf = ();

## NT Port
## exapnd all files for all partitions
        if ($params{'tab_'.$table.'_load_filf'} =~
/alltsdatafiles/ )
        {   if ($params{'tab_'.$table.'_part_ts'} =~
/^(.*)\#$/ && $dgp == 1)
            {
                $curts = "";
                $params{'tab_'.$table.'_part_ts'} =~
/^(.*)\#$/;
                $tsnam = $1;
                $params{'tab_'.$table.'_load_filf'} =
"";
                $numts =
$params{'tab_'.$table.'_#part'};
                $numfil =
$params{$params{'tab_'.$table.'_ts'}.'_#files'};

                for ($j = 1; $j <= $numts; $j++)
                {
                    $tspnam =
sprintf("%s%d",$tsnam,$j);
                    if (defined
$params{$tspnam.'_areas'})
                    {
                        $tsareas =
$params{$tspnam.'_area'} . "," .
$params{$tspnam.'_areas'};
                    }
                    else
                    {
                        $tsareas =
$params{$tspnam.'_area'};
                        for ($i = 1; $i < $numfil;
$i++)
                        {
                            $tsareas = $tsareas . "," .
$params{$tspnam.'_area'};
```

```
                    }
                }
                $tsarea[$j] = [ split(/,/,
$tsareas) ];
            }

            for ($i = 1; $i <= $numfil; $i++)
            {
                for ($j = 1; $j <= $numts; $j++)
                {
                    $tspnam =
sprintf("%s%d",$tsnam,$j);
                    (($i == $numfil) && ($j ==
$numts)) ?
                        ($addcomma = "") :
($addcomma = ",");
                    if ($tsarea[$j][$i-1] !~
/\\\\.\\/)
                    {
                      $nextfile =
sprintf("%s%s_%d.dbf%s",$tsarea[$j][$i-
1],$tspnam,$i,$addcomma);
                    }
                    else
                    {
# Temporay walkaround for 804 & SQLLDR bugs
#                       $nextfile =
sprintf("%s%s_%d%s",$tsarea[$j][$i-
1],$tspnam,$i,$addcomma);
                        if (defined
$params{'tab_'.$table.'_part_lfn'})
                          {
                            $nextfile =
sprintf("\\\\\\\\.\\\\%s%d_%d%s",$params{'tab_'.$table
.'_part_lfn'},$j,$i,$addcomma);
                          }
                        else
                        {
                          $nextfile =
sprintf("\\\\\\\\.\\\\%s_%d%s",$tspnam,$i,$addcomma);
                        }
                    }

$params{'tab_'.$table.'_load_filf'} .= $nextfile;
                }
             }
          }
          else
          {
             $params{'tab_'.$table.'_load_filf'} =
$params{$curts.'_datafiles'};
          }
       }
       elsif ($params{'tab_'.$table.'_load_filf'} =~
/^(.*)\#$/ )
          {
             $curts = "";
             $params{'tab_'.$table.'_load_filf'} =~
/^(.*)\#$/;
             $tsnam = $1;
             $params{'tab_'.$table.'_load_filf'} =
"";
             $tsareas = $params{$tsnam.'area'} . ","
. $params{$tsnam.'areas'};
              @tsarea = split(/,/, $tsareas);
             $numfil = @tsarea;
              @dfile = split(/,/,
$params{$tsnam.'datafiles'});
              $numfil = @dfile if ($numfil <
@dfile);
             for ($i = 0; $i < $numfil; $i++)
             {
                ($i == $numfil - 1) ? ($addcomma =
"") : ($addcomma = ",");
                if (@tsarea[0] !~ /\\\\.\\/)
                {
                  $nextfile =
sprintf("%s%s.dbf%s",@tsarea[$i],$dfile[$i],$addcomma)
;
                }
                else
                {
# Temporay workaround of 804 and SQLLDR bugs
#                   $nextfile =
sprintf("%s%s%s",@tsarea[$i],$dfile[$i],$addcomma);
                    $nextfile =
sprintf("\\\\\\\\.\\\\%s%s",$dfile[$i],$addcomma);

                }
                $params{'tab_'.$table.'_load_filf'}
.= $nextfile;
                }
            }

## NT End

        @filf = split(/,/,
$params{'tab_'.$table.'_load_filf'}) if defined
$params{'tab_'.$table.'_load_filf'};
        # relax the requirement a little bit by
allowing round robining
        $numfil = @filf;
## NT Port (fix the case for partition tables which
has only one file in each partition
##
##--
##      if (($numfil > 1) && ($numfil < $dp) && (($dp
% $numfil) == 0)) {
##          $p = $dp / $numfil;
##--
        if ((($numfil > 1) && ($numfil < $dp) && (($dp
% $numfil) == 0)) ||
                (($dgp == 1) && ($dp == 1)))
        {
            $p = 1;
            $p = ($dp / $numfil) if ($dp > 1);
## NT End
            $fil = "";
            for ($i = 0; $i < $p; $i++)
            {
                if ($i == 0) {
                    $fil =
$params{'tab_'.$table.'_load_filf'};
                } else {
                    $fil =
join(',',$fil,$params{'tab_'.$table.'_load_filf'});
                }
            }
            @filf = split(/,/, $fil);
        }

        $err = "fil" if ((@filf > 1) && ($dp != @filf)
&& ($dp != 1));
        $ffexist = 1 if @filf > 0;
        if ($err)
        {
            print "Error with load parameters for
table $table\n";
            print "  degree is $dp, $err list has bad
number of elements\n";
            return;
        }
        $usels = 1 if ((($dp > 1) && (@datf == 1) &&
$params{'tab_'.$table.'_load_datf'} !~ /\#/));

# expand all of the xxxf arrays to arrays of size $dp

        @ff = ();

        $dp = @filf if (($dp == 1) && ($dgp == 1));

        for ($i=0; $i < $dp; $i++)
        {
            $iplus = $i + 1;
            if (@ctlf > 1)
            {
                @cf[$i] = @ctlf[$i];
            }
            else
            {
                @cf[$i] = @ctlf[0];
                @cf[$i] =~ s/\#/$iplus/g;
            }
            if (@logf > 1)
            {
                @lf[$i] = @logf[$i];
            }
            else
            {
                @lf[$i] = @logf[0];
                @lf[$i] =~ s/\#/$iplus/g;
            }
            if (@badf > 1)
            {
                @bf[$i] = @badf[$i];
            }
            else
            {
                @bf[$i] = @badf[0];
                @bf[$i] =~ s/\#/$iplus/g;
```

```
            }
            if (@datf > 1)
            {
                @daf[$i] = @datf[$i];
            }
            else
            {
                # if load_degpar is larger than the
number partitions
                # and we are using pipes,
                # then load_degpar has to be a multiple
of the number
                # of partition and
load_degpar/#partitions sets of
                # pipes will be setup.
                # The goal is to maintain only one
pipe/flatfile per
                # sqlldr.

                if (($params{'tab_'.$table.'_#part'} >
0) &&
                    ($dp >
$params{'tab_'.$table.'_#part'})) {
                    if ($dp %
$params{'tab_'.$table.'_#part'} == 0) {
                        if (@datf[0] =~ /\#.*\#/) {
                            # set number

$snum=int($i/$params{'tab_'.$table.'_#part'}) + 1;
                            # partiton number

$pnum=$i%$params{'tab_'.$table.'_#part'} + 1;
                            @daf[$i] = @datf[0];
                            @daf[$i] =~ s/\#/$snum/;
                            @daf[$i] =~ s/\#/$pnum/;
                        } else {
                            @daf[$i] = @datf[0];
                            @daf[$i] =~ s/\#/$iplus/g;
                        }
                    }
                    else
                    {
                        print "Error: #partitons for
table $table does not divide load_degpar for the
table.\n";
                        print "degree is $dp, number of
partitions is $params{'tab_'.$table.'_#part'}.\n";
                        return;
                    }
                }
                else
                {
                    # assign
                    @daf[$i] = @datf[0];
                    @daf[$i] =~ s/\#/$iplus/g;
                }
            }
            if (@disf > 1)
            {
                @dif[$i] = @disf[$i];
            }
            else
            {
                @dif[$i] = @disf[0];
                @dif[$i] =~ s/\#/$iplus/g;
            }
            if (@filf > 1)
            {
                @ff[$i] = @filf[$i];
            }
            else
            {
                # round robining - temp fix
                @ff[$i] = @filf[0] if (@filf);
                if (defined
$params{$params{'tab_'.$table.'_ts'}.'_#files'}) {
                    $numfil =
$params{$params{'tab_'.$table.'_ts'}.'_#files'};
#MP                 print "$table \n";
#MP                 print
"$params{$params{'tab_'.$table.'_ts'}.'_#files'}\n";
                    $filnum = $i % $numfil + 1;
                    @ff[$i] =~ s/\#/$filnum/g;
                } else {
                    @ff[$i] =~ s/\#/$iplus/g;
                }
            }
        }

        $control = sprintf ("%s%s",
$params{'load_controlfile_area'}, @cf[0]);
        $control =~ s/\?/$ENV{'ORACLE_HOME'}/g;

        # add code to create head and tail of control
files

        $controlh = sprintf ("%s%s.head",
$params{'load_controlfile_area'}, $table);
        $controlh =~ s/\?/$ENV{'ORACLE_HOME'}/g;

        $controlt = sprintf ("%s%s.tail",
$params{'load_controlfile_area'}, $table);
        $controlt =~ s/\?/$ENV{'ORACLE_HOME'}/g;

        open (CTLFILEH, ">$controlh");
        open (CTLFILET, ">$controlt");
        print CTLFILEH "---\n";
        if (!($pre72) && defined
$params{'tab_'.$table.'_loadextent'})
            {
                print CTLFILEH "options\n";
                print CTLFILEH "(\n";
                print CTLFILEH "storage = (initial
$params{'tab_'.$table.'_loadextent'} next
$params{'tab_'.$table.'_loadextent'})\n";
                print CTLFILEH ")\n";
            }

#       print CTLFILE "unrecoverable\n" if
($params{'load_unrecoverable'} =~ /[tT][rR][uU][eE]/);
        print CTLFILEH "unrecoverable\n" if
($params{'load_unrecoverable'} =~ /[tT][rR][uU][eE]/);

        print CTLFILEH "load\n";
        print CTLFILEH "-- This is where INFILE should
go.\n";
#       print CTLFILEH
"$params{'load_insert_type'}\n";
        print CTLFILEH "into table $table\n";


        print CTLFILET "-- This is where PARTITION is
specified.\n";
        print CTLFILET
"$params{'load_insert_type'}\n";
        print CTLFILET "fields terminated by
$params{'load_field_terminator'}\n" if ($ud == 1);
        print CTLFILET "(\n";

        @tab_collist = split(/,/,
$params{'tab_'.$table.'_columns'});
        while ($col = shift(@tab_collist))
        {
            (@tab_collist == 0) ? ($addcomma = "") :
($addcomma = ",");
            $pos = "";
            $pos = sprintf ("position
(%s)",$params{'tab_'.$table.'_'.$col.'_pos'}) if ($ud
== 0);
            $ctlline = sprintf ("  %-20s %s
%s$addcomma", $col, $pos,
$params{'tab_'.$table.'_'.$col.'_loadcolx'});
            print CTLFILET "$ctlline\n";
        }
        print CTLFILET ")\n";
        close (CTLFILEH);
        close (CTLFILET);

# create the controlfiles (fixed-length fields or
delimited records)
        if ($params{'skip_mk_ldctlfs'} !~ /$table/)
        {
            #print "control $control\n";
            open (CTLFILE, ">$control");
            print CTLFILE "---\n";
            print CTLFILE "---  $table.ctl for
delimited records\n" if ($ud == 1);
            print CTLFILE "---  $table.ctl for fixed-
length fields\n" if ($ud == 0);
            print CTLFILE "---\n\n";
            if (!($pre72) && defined
$params{'tab_'.$table.'_loadextent'})
            {
                print CTLFILE "options\n";
                print CTLFILE "(\n";
                print CTLFILE "storage = (initial
$params{'tab_'.$table.'_loadextent'} next
$params{'tab_'.$table.'_loadextent'})\n";
                print CTLFILE ")\n";
            }
```

```
       print CTLFILE "unrecoverable\n" if
($params{'load_unrecoverable'} =~ /[tT][rR][uU][eE]/);
            print CTLFILE "load\n";
#           print CTLFILE
"$params{'load_insert_type'}\n";
            print CTLFILE "into table $table\n";

            print CTLFILE
"$params{'load_insert_type'}\n";
            print CTLFILE "fields terminated by
$params{'load_field_terminator'}\n" if ($ud == 1);
            print CTLFILE "(\n";
            @tab_collist = split(/,/,
$params{'tab_'.$table.'_columns'});
            while ($col = shift(@tab_collist))
            {
                (@tab_collist == 0) ? ($addcomma = "")
: ($addcomma = ",");
                $pos = "";
                $pos = sprintf ("position
(%s)",$params{'tab_'.$table.'_'.$col.'_pos'}) if ($ud
== 0);
                $ctlline = sprintf ("  %-20s %s
%s$addcomma", $col, $pos,
$params{'tab_'.$table.'_'.$col.'_loadcolx'});
                print CTLFILE "$ctlline\n";
            }
            print CTLFILE ")\n";
            close (CTLFILE);
        }

# create all of the "executable" load sections
# These variables will hold for all loaders for a
given table
        $upwd = "";
        $parbool = "";
        $dirbool = "";
        $silent = "";
        $dismax = "";
       $file = "";
        $errors = "";
        $rows = "";
        $bsize = "";
        $load = "";
        $skip = "";
       $dbgopt = "";
       $dbgtab = "";

       if ($up) {
            if ($table =~ /lineitem/)
            {
                $dbgtab = "L";
            }
            elsif ($table =~ /orders/)
            {
                $dbgtab = "O";
            }
            elsif ($table =~ /partsupp/)
            {
                $dbgtab = "S";
            }
            elsif ($table =~ /part/)
            {
                $dbgtab = "P";
            }
            elsif ($table =~ /customer/)
            {
                $dbgtab = "c";
            }
            elsif ($table =~ /supplier/)
            {
                $dbgtab = "s";
            }
            elsif ($table =~ /nation/)
            {
                $dbgtab = "n";
            }
            elsif ($table =~ /region/)
            {
                $dbgtab = "r";
            }
            $dbgopt .= "-T ".$dbgtab." ";
            $dbgopt .= "-s ". $params{'scale_factor'}
. " ";
            $dbgopt .= "-C ";
            $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?
                        $params{'load_dbgen_'.$table.'_
option_C'} :
                        $params{'load_dbgen_def_option_
C'});
            if ($dgp) {
                $dbgopt .= " -p -i ";
                $dbgopt .= (defined
$params{'load_dbgen_'.$table.'_input_params'}) ?
$params{'load_dbgen_'.$table.'_input_params'} :
$params{'load_dbgen_def_input_params'};
            }
        }

        $upwd = "$params{'user'}/$params{'passwd'}";
        $parbool = " parallel=true" if
(($params{'tab_'.$table.'_load_parallel'} =~ /true/)
|| ($dp > 1));
        $dirbool = " direct=true" if
(($params{'tab_'.$table.'_load_direct'} =~
/[Tt][Rr][Uu][Ee]/) || $parbool);
        $silent = "
silent=$params{'tab_'.$table.'_load_silent'}" if
defined $params{'tab_'.$table.'_load_silent'};
        $dismax = "
discardmax=$params{'tab_'.$table.'_load_dismax'}" if
defined $params{'tab_'.$table.'_load_dismax'};
        $errors = "
errors=$params{'tab_'.$table.'_load_errors'}" if
defined $params{'tab_'.$table.'_load_errors'};
        $rows = "
rows=$params{'tab_'.$table.'_load_rows'}" if defined
$params{'tab_'.$table.'_load_rows'};
        $bsize = "
bindsize=$params{'tab_'.$table.'_load_bsize'}" if
defined $params{'tab_'.$table.'_load_bsize'};

        if ($usels)
        {
            $split =
int($params{'tab_'.$table.'_#rows'} / $dp);
            $extra =
int($params{'tab_'.$table.'_#rows'} % $dp);
            $loadval = $split;
            $skipval = 0;
        }

## NT Port (use dbgen to create flat files)
##
##      # if using pipes, create the pipes
##
##      if($up)
##      {
##          for ($i=0;$i<$dp;$i++) {
##              $cmd = sprintf("%s%s
p\n",$params{'load_flatfile_area'},@daf[$i]);
##              &dump1("*mknod");
##          }
##          &dump0("*wait");
##
##          # start the dbgens
##
##          if ($dgp) {
##              $nset = int($dp /
$params{'tab_'.$table.'_#part'});
##              $nperset = int($dbgpar / $nset);
##          }
##
##          for ($i=0;$i<$dbgpar;$i++) {
##              $cmd = $params{'load_flatfile_area'} .
"\n";
##              $cmd2 = sprintf("%s -S %d", $dbgopt,
($i + 1));
##              if ($dgp) {
##                  # if the output prefix name is
specified with #
##                  # divide output file names into
$dp/#numpart sets
##                  $fpre = (defined
$params{'load_dbgen_'. $table.'_output_prefix'}) ?
$params{'load_dbgen_'.$table.'_output_prefix'} :
$params{'load_dbgen_def_output_prefix'};
##                  if ($fpre =~ /\#/) {
##                      $snum = int($i/$nperset) + 1;
##                      $fpre =~ s/\#/$snum/g;
##                  }
##                  $cmd2 .= sprintf(" -o %s ", $fpre);
##              }
##
### Added 4/28?  - Ari
### need to force in case pipe still there, and need
to kick
### off in the backgroud from here
##              $cmd2 = " -f " . $cmd2;
```

```
##             $cmd2 .= " & ";
##
##             $cmd2 .= "\n";
##             &dump2("*dbgen");
##         }
##     }
##
##     &dump0("*wait");
## NT End

        for ($i=0; $i < $dp; $i++)
        {
            &advmulti();
            $control = "";
            $log = "";
            $bad = "";
            $data="";
            $discard = "";
            $control = sprintf (" control=%s%s",
$params{'load_controlfile_area'}, @cf[$i]);
            $control = sprintf (" control=/host%s%s",
$params{'load_controlfile_area'}, @cf[$i]) if
($params{'special_machine'} eq 'ncube');
            $log = sprintf (" log=%s%s",
$params{'load_otherfile_area'}, @lf[$i]) if ($lfexist
== 1);
            $bad = sprintf (" bad=%s%s",
$params{'load_otherfile_area'}, @bf[$i]) if ($bfexist
== 1);
            $data = sprintf (" data=%s%s",
$params{'load_flatfile_area'}, @daf[$i]) if
($dafexist== 1);
            $discard = sprintf (" discard=%s%s",
$params{'load_otherfile_area'}, @dif[$i]) if
($difexist== 1);
            if ($ffexist ==1)
            {
                if ($curts == "")
                {
                    $file = sprintf (" file=%s",
@ff[$i]);
                }
                else
                {
                    $file = sprintf (" file=%s%s",
$params{$curts.'_area'}, @ff[$i]);
                }
            }
## NT Port
## Change to UPPER case string as a workaround for
SQLLDR
            $file =~ tr/a-z/A-Z/;
## NT End
            if ($usels)
            {
                $loadval = $loadval + 1 if ($extra >
$i);
                $load = sprintf (" load=%d",
$loadval);
                $skip = sprintf (" skip=%d",
$skipval);
                $skipval = $skipval + $loadval;
                $loadval = $split;
            }
##          if ($up)
##          {
##              $load = " load=99999999";
##          }

            $cmd = sprintf
("%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s\n",$upwd,$control,$
log,$bad,$data,$discard,$dismax,$skip,$load,$errors,$r
ows,$bsize,$silent,$dirbool,$parbool,$file);
            &dump1("*load");
            &dump0("*wait") if ((!$parbool) &&
(!defined $params{'load_no_waits'}));
        }
        &dump0("*wait") if !defined
$params{'load_no_waits'};

##--    if($up)
##      {
##          for ($i=0;$i<$dp;$i++) {
##              $cmd = sprintf("rm -f
%s%s\n",$params{'load_flatfile_area'},@daf[$i]);
##              &dump1("*sh");
##          }
##          &dump0("*wait");
##--    }
        &time0("End $table load");
    }

    $cmd = "shutdown\n"; #^^sd
    &dump1("*sql");
    &dump0("*wait");
    &time0("Done data population");
}


sub ixcre
{

&dump0("#############################################
##############################");
    &dump0("# Index Creation Phase");
    &time0("Begin index creation");
    foreach $index (@indexlist)
    {
        if (defined $params{'ind_'.$index.'_table'})
        {
            $ob_type = 'table';
            $ob_stype = 'tab';
        }
        else
        {
            $ob_type = 'view';
            $ob_stype = 'view';
        }

        &time0("Begin creating index $index");
        &advmulti();
        $cmd .= "connect
$params{'user'}/$params{'passwd'};\n";
        $cmd .= "drop index $index;\n";
        $cmd .= "create ";
        if ($params{'ind_'.$index.'_unique'} =~
/[tT][rR][uU][eE]/)
        {
            $cmd .= "unique ";
        } elsif ($params{'ind_'.$index.'_bitmap'} =~
/[tT][rR][uU][eE]/) {
            $cmd .= "bitmap ";
        }
        $cmd .= "index $index\n";
        if (defined
$params{'ind_'.$index.'_'.$ob_type})
        {
            $cmd .= "on
$params{'ind_'.$index.'_'.$ob_type}
($params{'ind_'.$index.'_'.$ob_stype.'cols'})\n";
            $indtab =
$params{'ind_'.$index.'_'.$ob_type};
        }
        else
        {
            $cmd .= "on cluster
$params{'ind_'.$index.'_cluster'}\n";
        }
        $cmd .= "pctfree
$params{'ind_'.$index.'_%f'}\n" if defined
$params{'ind_'.$index.'_%f'};
        $cmd .= "initrans
$params{'ind_'.$index.'_it'}\n" if defined
$params{'ind_'.$index.'_it'};
        $cmd .= "maxtrans
$params{'ind_'.$index.'_mt'}\n" if defined
$params{'ind_'.$index.'_mt'};
        if ($params{'compatible'} =~ /7\./)
        {
            $cmd .= "unrecoverable\n" if
($params{'ind_'.$index.'_unrecoverable'} =~
/[tT][rR][uU][eE]/);
        } elsif ($params{'compatible'} =~ /8\./) {
            $cmd .= "nologging\n" if
($params{'ind_'.$index.'_nolg'} =~
/[tT][rR][uU][eE]/);
        }
        $cmd .= "compute statistics\n" if
($params{'ind_'.$index.'_compstats'} =~
/[tT][rR][uU][eE]/);
        $cmd .= "tablespace
$params{'ind_'.$index.'_ts'}\n" if ((defined
$params{'ind_'.$index.'_ts'}) && ($params{'skip_ts'}
!~ /index/));
        $cmd .= "storage
$params{'ind_'.$index.'_storage'}\n" if defined
$params{'ind_'.$index.'_storage'};
        $cmd .= "reverse\n" if
($params{'ind_'.$index.'_reverse'} =~
```

```
/[tT][rR][uU][eE]/);
        $cmd .= "nosort\n" if
($params{'ind_'.$index.'_nosort'} =~
/[tT][rR][uU][eE]/);

        if ((defined $params{'ind_'.$index.'_pardeg'})
|| (defined $params{'ind-'.$index.'_parinst'}))
        {
            $cmd .= "parallel";
            if ($params{'ind_'.$index.'_pardeg'}
=~/[dD][eE][fF][aA][uU][lL]/)
            {
                $cmd .= "\n";
            }
            else
            {
                $cmd .= " (degree
$params{'ind_'.$index.'_pardeg'} "
                    if defined
$params{'ind_'.$index.'_pardeg'};
                $cmd .= "instances
$params{'ind_'.$index.'_parinst'}"
                    if defined
$params{'ind_'.$index.'_parinst'};
                $cmd = $cmd . ")\n";
            }
        }

        # deal with partitioning

        if ($params{'ind_'.$index.'_partition'} =~
/[lL][oO][cC][aA][lL]/)
        {
            $numpart =
$params{$ob_stype.'_'.$indtab.'_#part'};
            $cmd .= "local";
            if (defined
$params{'ind_'.$indtab.'_partnames'})
            {
                &exp_ind_part_l;
            }
            else
            {
                $cmd .= "\n";
            }
        } elsif ($params{'ind_'.$index.'_partition'}
=~ /[gG][lL][oO][bB][aA][lL]/) {
            $numpart =
$params{'ind_'.$index.'_#part'};
            $cmd .= "global partition by range (";
            &exp_ind_part_g;
        }
        $cmd .= ";\n";
        &dump1("*sql");
    }


# Constraints which are enabled after the load
    foreach $const (@constlist)
    {
        &time0("Begin creating constraint $const");
        &advmulti();

        $cmd .= "connect
$params{'user'}/$params{'passwd'};\n";
        if ($params{'con_'.$const.'_constraint'} =~
/null/)
        {
            @collist = split(/,/,
$params{'con_'.$const.'_columns'});
            $cmd .= "alter table
$params{'con_'.$const.'_table'} ";
            $cmd .= "modify (";
            while ($col = shift(@collist))
            {
                (@collist == 0) ? ($addcomma = "") :
($addcomma = ",");
                $cmd .= "$col
$params{'con_'.$const.'_constraint'}$addcomma";
            }
            $cmd .= ");\n";
        }
        else
        {
            $cmd .= "alter table
$params{'con_'.$const.'_table'} drop constraint
$const;\n";
            $cmd .= "alter table
$params{'con_'.$const.'_table'} ";
            $cmd .= "add constraint $const\n";
            $cmd .= "
$params{'con_'.$const.'_constraint'} key
($params{'con_'.$const.'_columns'}) ";
            if ($params{'con_'.$const.'_has_index'} =~
/[Tt][Rr][Uu][Ee]/) {
                $cmd .= "using index";
            }
            if ($params{'con_'.$const.'_disable'} =~
/[Tt][Rr][Uu][Ee]/) {
                $cmd .= "disable;\n";
            } else {
                $cmd .= ";\n";
            }
            if ($params{'con_'.$const.'_constraint'}
=~ /foreign/) {
                $cmd .= "alter table
$params{'con_'.$const.'_table'} ";
                $cmd .= "enable novalidate primary
key;\n";
            }
            if ($params{'con_'.$const.'_constraint'}
=~ /foreign/) {
                $cmd .= "alter table
$params{'con_'.$const.'_table'} ";
                $cmd .= "enable primary key";
            }
            if ($params{'con_'.$const.'_has_index'} =~
/[Tt][Rr][Uu][Ee]/) {
                $cmd .= "";
            } else {
                $cmd .= "using index\n";
                $cmd .= "pctfree
$params{'con_'.$const.'_%f'}\n" if defined
$params{'con_'.$const.'_%f'};
                $cmd .= "initrans
$params{'con_'.$const.'_it'}\n" if defined
$params{'con_'.$const.'_it'};
                $cmd .= "maxtrans
$params{'con_'.$const.'_mt'}\n" if defined
$params{'con_'.$const.'_mt'};
                $cmd .= "tablespace
$params{'con_'.$const.'_ts'}\n" if ((defined
$params{'con_'.$const.'_ts'}) && ($params{'skip_ts'}
!~ /index/));
                $cmd .= "storage
$params{'con_'.$const.'_storage'}\n" if defined
$params{'con_'.$const.'_storage'};
                $cmd .= "nosort\n" if
($params{'con_'.$const.'_nosort'} =~ /true/);
                $cmd .= ";\n";
            }
        }
        &dump1("*sql");
        &dump0("*wait") if (($os cmp "nt") ==0);
        &time0("Done creating constraint $const");
    }
# Alter DOP of indexes (NT support)
    if (($os cmp "nt") ==0)
    {
        &dump0("*wait");
        &time0("Alter DOP of indexes");
        $cmd  = "connect
$params{'user'}/$params{'passwd'};\n";

        foreach $index (@indexlist)
        {
            &advmulti();

            if (defined
$params{'ind_'.$index.'_pardeg_alter'})
            {
                $cmd .= "alter index $index parallel";
                $cmd .= " (degree
$params{'ind_'.$index.'_pardeg_alter'}";
                $cmd .= " instances
$params{'ind_'.$index.'_parinst'});\n";
            }
        }
        &dump1("*sql");
        &dump0("*wait");
        &time0("Done altering DOP of indexes");
        &time0("Done index creation");
    }
} # end of ixcre


sub exp_ind_part_l
{
    @ind_partnames = ();
    $params{'ind_'.$index.'_#part'} =
```

```
        $params{$ob_stype.'_'.$params{'ind_'.$index.'_
'.$ob_type}.'_#part'};

    if (!defined $params{'ind_'.$index.'_partnames'})
    {
         # if no partnames are specified, use a default
part name
        for ($i = 1; $i <=
$params{'ind_'.$index.'_#part'}; $i++)
        {
            ($i == $params{'ind_'.$index.'_#part'}) ?
($addcomma = "") :
                ($addcomma = ",");
             $nextfile = sprintf("%s%s%d%s",
$index,"_p", $i, $addcomma );
             $params{'ind_'.$index.'_partnames'} =
                 $params{'ind_'.$index.'_partnames'} .
$nextfile;
        }
    }

    @ind_partnames =
split(/,/,$params{'ind_'.$index.'_partnames'});
    if ($ind_partnames[0] =~ /\#/)
    {
        $filenm = shift(@ind_partnames);
        $savename = $filenm;
        $params{'ind_'.$index.'_partnames'} = "";
        # indtab defined in ixcre
        for ($i = 1; $i <= $numpart; $i++)
        {
            $filenm =~ s/\#/$i/g;
            ($i == $numpart) ? ($addcomma = "") :
($addcomma = ",");
            $params{'ind_'.$index.'_partnames'} =
                $params{'ind_'.$index.'_partnames'} .
$filenm . $addcomma;
            $filenm = $savename;
        }
        @ind_partnames =
split(/,/,$params{'ind_'.$index.'_partnames'});
        print "Expanded $savename
to:\n$params{'ind_'.$index.'_partnames'}\n\n" if
$verbose;
    } else {
        if (@ind_partnames != $numpart)
        {
            print "number of partitions $numpart from
the base table $indtab\ndoes not equal to the number
of partition names defined in
ind_".$index."_partnames.\n";
            exit(-1);
        }
    }

    &process_ind_part_ts;

    # complete the local partition index statement
    if ($numpart >0)
    {
        $cmd .= "(\n";
        for ($i=0; $i < $numpart; $i++)
        {
            $cmd .= "partition ";
            $cmd .= $ind_partnames[$i] . "\n" if
@ind_partnames;

&process_part_params('ind','%f','pctfree',$index,
                                (@ind_partnames) ?
$ind_partnames[$i] : "");

&process_part_params('ind','%u','pctused',$index,
                                (@ind_partnames) ?
$ind_partnames[$i] : "");

&process_part_params('ind','it','initrans',$index,
                                (@ind_partnames) ?
$ind_partnames[$i] : "");

&process_part_params('ind','mt','maxtrans',$index,
                                (@ind_partnames) ?
$ind_partnames[$i] : "");
            if ((@ind_partnames) &&
                (defined
$params{'ind_'.$index.'_'.$ind_partnames[$i].'_ts'}))
            {
                $cmd .= 'tablespace '.

$params{'ind_'.$index.'_'.$ind_partnames[$i].'_ts'} .
"\n";

                &process_part_storage('ind',$index,$ind
_partnames[$i]);
            } elsif ((@ind_partnames) &&
                     defined
$params{'ind_'.$index.'_part_ts'})
            {
                $cmd .= 'tablespace ' .
$ind_part_ts[$i] . "\n";
                &process_part_storage('ind',$index,$ind
_partnames[$i]);
            }
            if (@ind_partnames &&
                defined
$params{'ind_'.$index.'_'.$ind_partnames[$i].'_nolg'})
            {
                $cmd .= "nologging\n"
                    if
($params{'ind_'.$index.'_'.$ind_partnames[$i].'_nolg'}
                        =~ /[tT][rR][uU][eE]/);
            } elsif (defined
$params{'ind_'.$table.'_part_def_nolg'}) {
                $cmd .= "nologging\n"
                    if
($params{'ind_'.$index.'_part_def_nolg'} =~
                        /[tT][rR][uU][eE]/);
            }
            $cmd .= (($i+1) == $numpart) ? "\n" :
",\n";
        }
        $cmd .= ")\n";
    }
} # end of exp_ind_part_l


sub exp_ind_part_g
{
    @ind_partnames = ();
    @pcollist = split(/,/,
$params{'ind_'.$index.'_partcol'});
    $cmd .= "$params{'ind_'.$index.'_partcol'})\n(\n";

    if (!defined $params{'ind_'.$index.'_partnames'})
    {
         # if no partnames are specified, use a default
part name
        for ($i = 1; $i <=
$params{'ind_'.$index.'_#part'}; $i++)
        {
            ($i == $params{'ind_'.$index.'_#part'}) ?
($addcomma = "") :
                ($addcomma = ",");
             $nextfile = sprintf("%s%s%d%s",
$index,"_p", $i, $addcomma );
             $params{'ind_'.$index.'_partnames'} =
                 $params{'ind_'.$index.'_partnames'} .
$nextfile;
        }
    }

    @ind_partnames =
split(/,/,$params{'ind_'.$index.'_partnames'});

    if ($ind_partnames[0] =~ /\#/)
    {
        $filenm = $ind_partnames[0];
        $savename = $filenm;
        $params{'ind_'.$index.'_partnames'} = "";
        # indtab defined in ixcre
        for ($i = 1; $i <= $numpart; $i++)
        {
            $filenm =~ s/\#/$i/g;
            ($i == $numpart) ? ($addcomma = "") :
($addcomma = ",");
            $params{'ind_'.$index.'_partnames'} =
                $params{'ind_'.$index.'_partnames'} .
$filenm . $addcomma;
            $filenm = $savename;
        }
        @ind_partnames =
split(/,/,$params{'ind_'.$index.'_partnames'});
        print "Expanded $savename
to:\n$params{'ind_'.$index.'_partnames'}\n\n" if
$verbose;
    } else {
        if (@ind_partnames != $numpart)
        {
            print "number of partitions $numpart from
the base table $indtab\ndoes not equal to the number
of partition names defined in
ind_$index_partnames.\n";
```

```
            exit(-1);
        }
    }

    # process boundaries and tablespaces

    &process_ind_part_brys;
    &process_ind_part_ts;

    # complete the local partition index statement

    for ($i=0; $i < $numpart; $i++)
    {
        $cmd .= "partition " . $ind_partnames[$i] . "
values less than ";
        if ($i==$params{'ind_'.$table.'_#part'}-1) {
          $cmd .= "(MAXVALUE)\n";
        }
        else {
          $cmd .= "(" . $ind_part_vals[$i] . ")\n";
        }

        &process_part_params('ind','%f','pctfree',$ind
ex,$ind_partnames[$i]);
        &process_part_params('ind','%u','pctused',$ind
ex,$ind_partnames[$i]);

&process_part_params('ind','it','initrans',$index,$ind
_partnames[$i]);

&process_part_params('ind','mt','maxtrans',$index,$ind
_partnames[$i]);

        if (defined
$params{'ind_'.$index.'_'.$ind_partnames[$i].'_ts'})
        {
            $cmd .= 'tablespace '.
                $params{'ind_'.$index.'_'.$ind_partname
s[$i].'_ts'} . "\n";

&process_part_storage('ind',$index,$ind_partnames[$i])
;
        } elsif (defined
$params{'ind_'.$index.'_part_ts'})
        {
            $cmd .= 'tablespace ' . $ind_part_ts[$i] .
"\n";

&process_part_storage('ind',$index,$ind_partnames[$i])
;
        }
        if (defined
$params{'ind_'.$index.'_'.$ind_partnames[$i].'_nolg'})
        {
            $cmd .= "nologging\n"
                if
($params{'ind_'.$index.'_'.$ind_partnames[$i].'_nolg'}
                    =~ /[tT][rR][uU][eE]/);
        } elsif (defined
$params{'ind_'.$table.'_part_def_nolg'}) {
            $cmd .= "nologging\n"
                if
($params{'ind_'.$index.'_part_def_nolg'} =~
                    /[tT][rR][uU][eE]/);
        }
        $cmd .= (($i+1) == $numpart) ? "" : ",\n";
    }
    $cmd .= ")\n";

} # end of exp_ind_part_g


sub process_ind_part_ts
{
    # is ind_<name>_part_ts is in the form of XXXX#,
expand it
    # else treat it as a comma separated list of ts
names
    if (defined $params{'ind_'.$index.'_part_ts'})
    {
        @ind_part_ts =
split(/,/,$params{'ind_'.$index.'_part_ts'});
        if ($ind_part_ts[0] =~ /\#/)
        {
            $filenm = shift(@ind_part_ts);
            $savename = $filenm;
            $params{'ind_'.$index.'_part_ts'} = "";
            for ($i = 1; $i <= $numpart; $i++)
            {
                $filenm =~ s/\#/$i/g;
                ($i == $numpart) ? ($addcomma = "") :
($addcomma = ",");
                $params{'ind_'.$index.'_part_ts'} =
                    $params{'ind_'.$index.'_part_ts'}
. $filenm . $addcomma;
                $filenm = $savename;
            }
            @ind_part_ts =
split(/,/,$params{'ind_'.$index.'_part_ts'});
        } else {
            if (@ind_part_ts != $numpart)
            {
               $numpart = @ind_part_ts;
               if (($numpart % $numfil) == 0) {
                   $p = $numpart / $numfil;
                   $fil = "";
                   for ($i = 0; $i < $p; $i++)
                   {
                      if ($i == 0) {
                         $fil =
$params{'ind_'.$index.'_part_ts'};
                      } else {
                         $fil =
join(',',$fil,$params{'ind_'.$index.'_part_ts'});
                      }
                   }
                   @ind_part_ts = split(/,/, $fil);
               } else {
                   print "Number of partitions
$numpart for table $index\n doesn't match
ind_$index_part_ts parameter.\n";
                   exit (-1);
               }
            }
        }
    }
} # end of process_ind_part_ts

sub process_ind_part_brys
{
    $cnt = 0;
    @ind_part_vals = ();

    foreach $col (@pcollist)
    {
        # add quotes for character strings and dates

        if
(($params{$ob_stype.'_'.$indtab.'_'.$col.'_type'} =~
/[cC][hH][aA][rR]/) ||

($params{$ob_stype.'_'.$indtab.'_'.$col.'_type'} =~
/[dD][aA][tT][eE]/))
        {
            $addquote = "'";
        }

        @ind_part_col = ();

        # calculate the values for l_orderkey

        if ($col =~ /^l_orderkey$/)
        {
          $high_l_orderkey = $params{'scale_factor'} *
1500000 * 4;
          $pval = 1;
          if (defined
$params{$ob_stype.'_'.$table.'_#part'})
          {
            $numpart =
$params{$ob_stype.'_'.$table.'_#part'};
          }
          else
          {
            $numpart =
$params{'ind_'.$index.'_#part'};
          }
          $incre = $high_l_orderkey / $numpart;
          for ($i=0; $i < $numpart-1; $i++) {
             $pval = $pval + $incre;
             push(@ind_part_col, $pval);
          }
          push(@ind_part_col, 'MAXVALUE');
        }
        else {
          if ($col =~ /^l_partkey$/) {
            $high_l_partkey = $params{'scale_factor'}
* 200000;
            $pval = 1;
            if (defined
```

```
$params{$ob_stype.'_'.$table.'_#part'})
            {
              $numpart =
$params{$ob_stype.'_'.$table.'_#part'};
            }
            else
            {
              $numpart =
$params{'ind_'.$index.'_#part'};
            }
            $incre = $high_l_partkey / $numpart;
            for ($i=0; $i < $numpart-1; $i++) {
               $pval = $pval + $incre;
               push(@ind_part_col, $pval);
            }
           push(@ind_part_col, 'MAXVALUE');
          }
          else
          {
            @ind_part_col =
split(/,/,$params{'ind_'.$index.'_'.$col.'_partvals'})
;
          }
        }

        if (@ind_part_col !=
$params{'ind_'.$index.'_#part'})
        {
            printf "Number of partition boundary
values %d for column $col in global index $index
doesn't match the number of partitions
($params{'ind_'.$index.'_#part'}) of the index\n",
($#ind_part_col + 1);
            exit(-1);
        }
        for ($i=0; $i <
$params{'ind_'.$index.'_#part'}; $i++)
        {
            ($cnt == $#pcollist) ? ($addcomma = "") :
($addcomma = ",");
           if ($ind_part_col[$i] =~
/[Mm][Aa][Xx][Vv][Aa][Ll][Uu][Ee]/) {
              $addquote = "";
            }
           if
($params{$ob_stype.'_'.$indtab.'_'.$col.'_type'} =~
/[dD][aA][tT][eE]/)
            {
              if ($i ==
$params{'ind_'.$index.'_#part'} - 1)
              {
                $ind_part_vals[$i] .= "MAXVALUE";
              } else {
                $nls_format = (defined
$params{$ob_stype.'_'.$indtab.'_'.$col.'_date_format'}
) ?
$params{$ob_stype.'_'.$indtab.'_'.$col.'_date_format'}
: "YYYY-MM-DD";
                $ind_part_vals[$i] .=
"to_date('".$ind_part_col[$i]."', '";
                $ind_part_vals[$i] .=
$nls_format."')".$addcomma;
              }
            }
              elsif
(!(($params{$ob_stype.'_'.$indtab.'_'.$col.'_type'} =~
/[iI][nN][tT][eE][gG][eE][rR]/) ||
($params{$ob_stype.'_'.$indtab.'_'.$col.'_type'} =~
/[nN][uU][mM][bB][eE][rR]/)))
              {
              if ($index=~/l_ored/) {
                print "Before:$ind_part_vals[$i]";
              }
              $ind_part_vals[$i] =
$ind_part_vals[$i] . $addquote.
                $ind_part_col[$i] . $addquote .
$addcomma ;
              if ($index=~/l_ored/) {
                print "After:$ind_part_vals[$i]\n";
              }

            } else {
              $ind_part_vals[$i] = $ind_part_col[$i].
$addcomma ;
            }
        }
        $cnt++;
    }
} # end of process_ind_part_brys


sub anlyz
{

&dump0("#############################################
##############################");
    &dump0("# Analyze Phase");
    &time0("Begin anlyze");

    if ($params{'analyze_type'} =~ /via package
dbms.stats/)
    {
        &time0("Begin via package dbms.stats
analyzing");
        foreach $object (@anlyzlist)
        {
            $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
            $phead="anl_".$object;
            $type = $params{$phead.'_object_type'};
            $type = "table" if
($params{$phead.'_object_type'} =~ /view/);
            $type = "table" if
($params{$phead.'_object_type'} =~ /viewlog/);
            $objectname = $object;
            $objectname = "MLOG\\\$_".$object if
($params{$phead.'_object_type'} =~ /viewlog/);
            $cmd .= "execute
dbms_stats.$params{$phead.'_anl_type'}_".$type."_stats
('$params{'user'}'";
            $cmd .= " , estimate_percent =>
$params{$phead.'_percent'}" if (defined
$params{$phead.'_percent'});
            $cmd .= " , degree =>
$params{$phead.'_degree'}" if (defined
$params{$phead.'_degree'});
            $cmd .= " , granularity =>
'$params{$phead.'_granularity'}'" if (defined
$params{$phead.'_granularity'});
            $cmd .= " , block_sample =>
$params{$phead.'_block_sample'}" if (defined
$params{$phead.'_block_sample'});
            $cmd .= " , '".$objectname."'" if
!($object =~ /schema/);
            $cmd .= ");\n";
            &dump1("*sql");
             &dump0("*wait") if ($object =~ /schema/);
        }
        &time0("End per schema analyzing");
    }
    else
    {
        foreach $object (@anlyzlist)
        {
             &time0("Begin analyzing $object");
            &advmulti();
            if (($params{'anl_'.$object.'_type'} =~
/table/) &&
                ($params{'tab_'.$object.'_#part'} > 1))
{
                $tab_entry = 'tab_'.$object;
                if (!defined
$params{$tab_entry.'_partnames'})
                {
                    # if no partnames are specified,
use a default part name
                    for ($i = 1; $i <=
$params{$tab_entry.'_#part'}; $i++)
                    {
                        ($i ==
$params{$tab_entry.'_#part'}) ? ($addcomma = "") :
                            ($addcomma = ",");
                       $nextfile = sprintf("%s%s%d%s",
$table,"_p",
                                                $i,
$addcomma );
                       $params{$tab_entry.'_partnames'
} =
$params{$tab_entry.'_partnames'} . $nextfile;
                    }
                }
                @tab_partnames =
split(/,/,$params{$tab_entry.'_partnames'});
                # if partnames is specified as XXXX#,
then expand
                if ($tab_partnames[0] =~ /\#/)
                {
                    $filenm = shift(@tab_partnames);
                    $savename = $filenm;
```

```
                    $params{$tab_entry.'_partnames'} =
"";
                    for ($i = 1; $i <=
$params{$tab_entry.'_#part'}; $i++)
                    {
                        $filenm =~ s/\#/$i/g;
                        ($i ==
$params{$tab_entry.'_#part'}) ? ($addcomma = "") :
                                ($addcomma = ",");
                        $params{$tab_entry.'_partnames'
} =

$params{$tab_entry.'_partnames'} . $filenm .
$addcomma;
                        $filenm = $savename;
                    }
                    @tab_partnames =
split(/,/,$params{$tab_entry.'_partnames'});
                    printf("Expanded $savename
to:\n$params{$tab_entry.'_partnames'}\n\n") if
$verbose;
                } else {
                    if (@tab_partnames !=
$params{$tab_entry.'_#part'})
                    {
                        print "Number of partitions
$params{$tab_entry.'_#part'} for $table doesn't
match\n _partnames parameter for
$params{$tab_entry.'_partnames'}\n";
                        exit(-1);
                    }
                }
                @tab_partnames =
split(/,/,$params{'tab_'.$object.'_partnames'});
                foreach $partname (@tab_partnames)
                {
                    $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
                    $cmd .= "analyze
$params{'anl_'.$object.'_type'} $object ";
                    $cmd .= "partition ($partname) ";
                    if (defined
$params{'anl_'.$object.'_estimate'})
                    {
                        $cmd .= "estimate statistics
$params{'anl_'.$object.'_estimate'};\n";
                    }
                    else
                    {
                        $cmd .= "compute
statistics;\n";
                    }
                    &dump1("*sql");
                }
            } else {
                if (($params{'anl_'.$object.'_type'} =~
/index/) &&
                    (defined
$params{'ind_'.$object.'_partition'})) {
                    $ind_entry = 'ind_'.$object;
                    if
($params{'ind_'.$object.'_partition'} =~
/[lL][oO][cC][aA][lL]/) {
                        $params{$ind_entry.'_#part'} =

$params{'tab_'.$params{$ind_entry.'_table'}.'_#part'};
                    }
                    if (!defined
$params{$ind_entry.'_partnames'})
                    {
                        # if no partnames are
specified, use a default part name
                        for ($i = 1; $i <=
$params{$ind_entry.'_#part'}; $i++)
                        {
                            ($i ==
$params{$ind_entry.'_#part'}) ?
                                ($addcomma = "") :
($addcomma = ",");
                            $nextfile =
sprintf("%s%s%d%s",$object,"_p",
                                                $i,
$addcomma );

$params{$ind_entry.'_partnames'} =
                                $params{$ind_entry.'_par
tnames'} . $nextfile;
                        }
                    }
                    @ind_partnames =
split(/,/,$params{$ind_entry.'_partnames'});

                    # if partnames is specified as
XXXX#, then expand
                    if ($ind_partnames[0] =~ /\#/)
                    {
                        $filenm =
shift(@ind_partnames);
                        $savename = $filenm;
                        $params{$ind_entry.'_partnames'
} = "";

                        for ($i = 1; $i <=
$params{$ind_entry.'_#part'}; $i++)
                        {
                            $filenm =~ s/\#/$i/g;

($i==$params{$ind_entry.'_#part'}) ? ($addcomma = "")
:
                                ($addcomma = ",");

$params{$ind_entry.'_partnames'} =
                                $params{$ind_entry.'_par
tnames'} . $filenm .
                                    $addcomma;
                            $filenm = $savename;
                        }
                        @tab_partnames=split(/,/,$param
s{$ind_entry.'_partnames'});
                        printf("Expanded $savename
to:\n$params{$ind_entry.'_partnames'}\n\n") if
$verbose;
                    } else {
                        if (@ind_partnames !=
$params{$ind_entry.'_#part'})
                        {
                            print "Number of partitions
$params{$ind_entry.'_#part'} for $table doesn't
match\n _partnames parameter for
$params{$ind_entry.'_partnames'}\n";
                            exit(-1);
                        }
                    }
                    @ind_partnames =
split(/,/,$params{'ind_'.$object.'_partnames'});
                    foreach $partname (@ind_partnames)
                    {
                        $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
                        $cmd .= "analyze
$params{'anl_'.$object.'_type'} $object ";
                        $cmd .= "partition ($partname)
";
                        if (defined
$params{'anl_'.$object.'_estimate'})
                        {
                            $cmd .= "estimate
statistics $params{'anl_'.$object.'_estimate'};\n";
                        }
                        else
                        {
                            $cmd .= "compute
statistics;\n";
                        }
                        &dump1("*sql");
                    }
                } else {
                    $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
#       $cmd .= "analyze
$params{'anl_'.$object.'_type'} $object ";
#       $cmd .= "delete statistics;\n";
                    $cmd .= "analyze
$params{'anl_'.$object.'_type'} $object ";
                    if (defined
$params{'anl_'.$object.'_estimate'})
                    {
                        $cmd .= "estimate statistics
$params{'anl_'.$object.'_estimate'};\n";
                    }
                    else
                    {
                        $cmd .= "compute
statistics;\n";
                    }
                    &dump1("*sql");
                }
            }
            &time0("Done analyzing $object");
```

```
        }
        &time0("End anlyze");
    }
}

sub chdop_old
{

&dump0("##############################################
##############################");
    &dump0("# Make DOP Phase");

    if (@chdoplist) {
        $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
    }
    foreach $object (@chdoplist)
    {
       if ($params{'chdop_'.$object.'_type'} =~
/table/) {
         $cmd .= "alter table $object parallel (degree
$chdoplist{$object} instances 1);\n";
       }
       else {
         $cmd .= "alter index $object parallel (degree
$chdoplist{$object} instances 1);\n";
       }
    }
    foreach $object (@chstorlist)
    {
       if ($params{'chstor_'.$object.'_type'} =~
/table/) {
         $cmd .= "alter table $object storage (next
$chstorlist{$object} pctincrease 0);\n";
       }
       else {
         $cmd .= "alter index $object storage (next
$chstorlist{$object} pctincrease 0);\n";
       }
    }
    &dump1("*sql");
}

sub chob
{

&dump0("################################b############
##############################");
    &dump0("# Make CHOB Phase");

    if (@choblist) {
        $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
    }
  FLOOP:
    while (@choblist)
    {
        $object = shift(@choblist);
        if ((defined
$params{'chob_tab_'.$object.'_degpar'}) || (defined
$params{'chob_tab'.$object.'_storage'}))
        { $objecttype = 'table';  $objecttypes =
'tab';$objectn=$object;}
        elsif ((defined
$params{'chob_ind_'.$object.'_degpar'}) || (defined
$params{'chob_ind'.$object.'_storage'}))
        {$objecttype = 'index'; $objecttypes =
'ind';$objectn=$object;}
        elsif ((defined
$params{'chob_viewlog_'.$object.'_degpar'}) ||
(defined $params{'chob_viewlog'.$object.'_storage'}))
        {$objecttype = 'table'; $objecttypes =
'viewlog';$objectn="MLOG\\\$_".$object}
        else
        {next FLOOP;}
        if (defined
$params{'chob_'.$objecttypes.'_'.$object.'_degpar'})
        {
            $cmd .= "alter $objecttype $objectn";
            if
($params{'chob_'.$objecttypes.'_'.$object.'_degpar'}
=~ /[dD][eE][fF][aA][uU][lL][tT]/)
            {
                $cmd .= " parallel";
            }
            elsif
($params{'chob_'.$objecttypes.'_'.$object.'_degpar'}
=~ /[nN][oO][pP][aA][rR][aA][lL][lL][eE][lL]/)
            {
                $cmd .= " noparallel";
            }
            else
            {
                $cmd .= " parallel
$params{'chob_'.$objecttypes.'_'.$object.'_degpar'}"
                    if (defined
$params{'chob_'.$objecttypes.'_'.$object.'_degpar'});
            }
            $cmd .= ";\n";
        }
        if (defined
$params{'chob_'.$objecttypes.'_'.$object.'_storage'})
        {
            $cmd .= "alter $objecttype $objectn";
            if (defined
$params{'chob_'.$objecttypes.'_'.$object.'_storage'})
            {
                $cmd .= " storage
($params{'chob_'.$objecttypes.'_'.$object.'_storage'})
;\n";
            }
        }
        if ($objecttypes eq 'tab')
        {
            delete
$params{'chob_tab_'.$object.'_degpar'};
            delete
$params{'chob_tab'.$object.'_storage'};
            push (@choblist,$object);
        }
    }
    &dump1("*sql");
}



sub expln
{

&dump0("##############################################
##############################");
    &dump0("# Explain Plan Phase");

    $params{'qry_explain_queries'} =
$params{'qry_all_queries'} if !defined
$params{'qry_explain_queries'};
    if ($params{'qry_explain_queries'})
    {
        $astr = "2_1";
        $astr = "2_".$params{'db_maxinstances'} if
defined $params{'db_maxinstances'};
        &alttabs($astr);
        @qrynames = split(/,/,
$params{'qry_explain_queries'});
        foreach $qname (@qrynames)
        {
            $cmd = "";
            $cmd .= "$params{'qry_area'}expl." .
$qname . "\n";
            $cmd2 = "";
            $cmd2 = $cmd2 . "connect
$params{'user'}/$params{'passwd'};\n";
            $cmd2 = $cmd2 . "set longwidth 360\n";
            if (defined
$params{'qry_'.$qname.'_text'})
            {
                $cmd2 = $cmd2 . "drop table
plan_table;\n";
                $cmd2 = $cmd2 . '@' .
"$params{'dd_sql_area'}" . "utlxplan.sql\n";
                $cmd2 = $cmd2 . "explain plan for\n";
                $cmd2 = $cmd2 .
"$params{'qry_'.$qname.'_text'}\n";
                $cmd2 = $cmd2 . "select level,
rpad(lpad(' ',2*(level-1)) || operation || ' ' ||
object_name || ' ' || options || ' (' || object_node
|| ',' || cost || ',' || cardinality || ')',68)\n";
                $cmd2 = $cmd2 . "from plan_table\n";
                $cmd2 = $cmd2 . "connect by
parent_id=prior id start with id=0;\n";
                $cmd2 = $cmd2 . "select level, lpad('
',2*(level-1)) || operation || ' ' || object_name || '
' || options || ' (' || object_node || ')',\n";
                $cmd2 = $cmd2 . "other_tag, other\n";
                $cmd2 = $cmd2 . "from plan_table\n";
                $cmd2 = $cmd2 . "connect by
parent_id=prior id start with id=0;\n";
            }
            else
```

```
            {
                if (defined
$params{'qry_'.$qname.'_texte'})
                {
                    @expln_texts = split(/,/,
$params{'qry_'.$qname.'_texte'});
                    $text_index = 1;
                    while ($expln_next = shift
(@expln_texts))
                    {
                      EXPLNLOOP:
                        while ($expln_next ne
('text'.$text_index))
                        {
                            if (!defined
$params{'qry_'.$qname.'_text'.$text_index})
                            {
                                print "\n** Missing
text for $qname: text$text_index\n";
                                last EXPLNLOOP;
                            }
                            $cmd2 = $cmd2 .
"$params{'qry_'.$qname.'_text'.$text_index}\n";
                            $text_index++;
                        }
                        $cmd2 = $cmd2 . "drop table
plan_table;\n";
                        $cmd2 = $cmd2 . '@' .
"$params{'dd_sql_area'}" . "utlxplan.sql\n";
                        $cmd2 = $cmd2 . "explain plan
for\n";
                        $sqltext =
'qry_'.$qname.'_'.$expln_next;
                        $cmd2 = $cmd2 .
"$params{$sqltext}\n";
                        $cmd2 = $cmd2 . "select level,
rpad(lpad(' ',2*(level-1)) || operation || ' ' ||
object_name || ' ' || options || ' (' || object_node
|| ',' || cost || ',' || cardinality || ')',68)\n";
                        $cmd2 = $cmd2 . "from
plan_table\n";
                        $cmd2 = $cmd2 . "connect by
parent_id=prior id start with id=0;\n";
                        $cmd2 = $cmd2 . "select level,
lpad(' ',2*(level-1)) || operation || ' ' ||
object_name || ' ' || options || ' (' || object_node
|| ')',\n";
                        $cmd2 = $cmd2 . "other_tag,
other\n";
                        $cmd2 = $cmd2 . "from
plan_table\n";
                        $cmd2 = $cmd2 . "connect by
parent_id=prior id start with id=0;\n";
                        $text_index++;
                    }
                    while (defined
$params{'qry_'.$qname.'_text'.$text_index})
                    {
                        $cmd2 = $cmd2 .
"$params{'qry_'.$qname.'_text'.$text_index}\n";
                        $text_index++;
                    }
                 }
                else
                {
                    print "\n** No sql text or texte
corresponding to $qname\n";
                }
            }
            &dump2("*sql2");
        }
    }
}

sub query
{

&dump0("#############################################
##############################");
    &dump0("# Query Phase");

    $params{'qry_run_queries'} =
$params{'qry_all_queries'} if !defined
$params{'qry_run_queries'};
    if ($params{'qry_run_queries'})
    {
        $params{'qry_config'} =  "1_1" if !defined
$params{'qry_config'};
        @qconfs = split(/,/, $params{'qry_config'});
        foreach $qconf (@qconfs)
        {
            &alttabs($qconf);
            if ($params{'qry_separate_queries'} !~
/true/)
            {
                $cmd = "";
                $cmd .= "$params{'qry_area'}qry.all."
. $qconf . "\n";
                $cmd2 = "";
                $cmd2 = $cmd2 . "set timing on\n";
                $cmd2 = $cmd2 . "connect
$params{'user'}/$params{'passwd'};\n";
            }
            @qrynames = split(/,/,
$params{'qry_run_queries'});
            foreach $qname (@qrynames)
            {
                if ($params{'qry_separate_queries'} =~
/true/)
                {
                    $cmd = "";
                    $cmd .= "$params{'qry_area'}qry."
. $qname . "." . $qconf . "\n";
                    $cmd2 = "";
                    $cmd2 = $cmd2 . "set timing on\n";
                    $cmd2 = $cmd2 . "connect
$params{'user'}/$params{'passwd'};\n";
                }
                for ($i = 0; $i <
$params{'qry_number_trials'}; $i++)
                {
                    if (defined
$params{'qry_'.$qname.'_text'})
                    {
                        $cmd2 = $cmd2 .
"$params{'qry_'.$qname.'_text'}\n";
                    }
                    else
                    {
                        $text_index = 1;
                      WTLOOP:
                        while(1)
                        {
                            if (defined
$params{'qry_'.$qname.'_text'.$text_index})
                            {
                                $cmd2 = $cmd2 .
"$params{'qry_'.$qname.'_text'.$text_index}\n";
                                $text_index++;
                            }
                            else
                            {
                                print "No sql text
corresponding to $qname\n" if ($text_index == 1);
                                last WTLOOP;
                            }
                        }
                    }
                }
                if ($params{'qry_separate_queries'} =~
/true/)
                {
                    &dump2("*sql2");
                }
            }
            if ($params{'qry_separate_queries'} !~
/true/)
            {
                &dump2("*sql2");
            }
        }
    }
}

#****************************************************
**********************
#****************************************************
**********************
# Utility Functions
#****************************************************
**********************
#****************************************************
**********************

sub prepmulti
{
# Some multi-instance stuff
    $num_inst = $params{$phase.'_num_inst'};
    if ($num_inst > 1)
    {
```

```
        $multi = 1;
        $cur_inst = 1;
        &advmulti(); # Make sure it's set to some
appropriate value
    }
    else
    {
        $multi = 0;
    }
}

sub advmulti
{
    if ($multi)
    {
        &setinst($cur_inst);
        $cur_inst = ($cur_inst % $num_inst) + 1;
    }
}

sub startdb #does not work for OPS systems!!
            #if you have an OPS system use startdb_old
{
    $initora=$params{'startupfile_'.$_[0]};
    $mounttype=$_[1];
    $cmd = "startup pfile=$initora $mounttype\n";
    &dump1("*sql");
}
sub startdb_old
{
# Get init.oras for startup/shutdown
    $checkios = 0;
    $ifile = sprintf ("%s%s", $params{'dbs_area'},
"init_$params{'oracle_sid'}.ora");
    $checkios = 1 if (!-e $ifile);
    for ($j = 1; $j <= $num_inst; $j++)
    {
        $iofname[$j] = sprintf ("%s%s",
$params{'dbs_area'},
"init".$j."_$params{'oracle_sid'}.ora");
        $iofname[$j] = sprintf ("/host%s",
$iofname[$j])
            if ($params{'special_machine'} eq
'ncube');
        $iofname[$j] =~ s/\?/$ENV{'ORACLE_HOME'}/g;
        $checkios = 1 if (!-e $iofname[$j]);
    }
    &createios() if $checkios;

# startup exclusive/shared single-instance/multi-
instance
    if ($multi)
    {
        for ($loop = 1; $loop <= $num_inst; $loop++)
        {
            &setinst($loop);
            $cmd = "startup shared
pfile=$iofname[$loop];\n";
            &dump1("*sql");
            &dump0("*wait") if
($params{'sync_startup'} =~ /true/);
        }
    }
    else
    {
        $cmd = "startup pfile=$iofname[1];\n";
        &dump1("*sql");
    }
    &dump0("*wait");
}

sub shutdb
{
# shutdown whatever might be running
    if ($multi)
    {
        for ($loop = $num_inst; $loop >= 1; $loop--)
        {
            &setinst($loop);
            $cmd = "shutdown
$params{'shutdown_level'};\n";
            &dump1("*sql");
        }
    }
    else
    {
        $cmd = "shutdown
$params{'shutdown_level'};\n";
        &dump1("*sql");
    }
    &dump0("*wait");
}

sub create_ts
{
    $tsname = shift(@_);      # 1. parameter is the
list of tablespaces

    &dump0("# creating $params{'user'}'s $tsname
tablespace");
    &advmulti();
    @ts_datafiles = split(/,/,
$params{$tsname.'_datafiles'});
    $ts_datafile = shift (@ts_datafiles);

    if ($params{$tsname.'_managed'} =~
/[uU][sS][eE][rR]/)
    {                         # user managed temporary
tablespace
        if ($params{$tsname.'_temporary'} =~
/[tT][rR][uU][eE]/)
            {                 # temporary tablespace
                $cmd .= "drop tablespace $tsname
including contents;\n";
                $cmd .= "create tablespace $tsname
temporary\n";
            }
            else              # user managed permanent
tablespace
            {
                $cmd .= "drop tablespace $tsname
including contents;\n";
                $cmd .= "create tablespace $tsname\n";
            }
        $cmd .= "datafile
'$params{$tsname.'_area'}$ts_datafile' size
$params{$tsname.'_first_size'}
$params{$tsname.'_options'}\n";
        $cmd .= "default storage
$params{$tsname.'_storage'}\n" if defined
$params{$tsname.'_storage'};
    }
    else                        # system managed
temporary tablespace
    {
        if ($params{$tsname.'_temporary'} =~
/[tT][rR][uU][eE]/)
            {                 # temporary tablespace
                $cmd .= "drop tablespace $tsname
including contents;\n";
                $cmd .= "create temporary tablespace
$tsname\n";
                $cmd .= "tempfile
'$params{$tsname.'_area'}$ts_datafile' size
$params{$tsname.'_first_size'}
$params{$tsname.'_options'}\n";
                $cmd .= "extent management local\n";
                if (defined
($params{$tsname.'_extent_size'}) &&
($params{$tsname.'_extent_size'} =~
/[aA][uU][Tt][oO][Aa][lL][lL][oO][cC][aA][tT][eE]/))
                {
                    $cmd .= "autoallocate\n";
                }
                elsif (defined
($params{$tsname.'_extent_size'}) &&
!($params{$tsname.'_extent_size'} =~
/[aA][uU][Tt][oO][Aa][lL][lL][oO][cC][aA][tT][eE]/))
                {
                    $cmd .= "uniform size
".$params{$tsname.'_extent_size'}."\n";
                }
                else
                {
                    $cmd .= "uniform";
                }
            }
        else            # system managed permanent
tablespace
        {
            $cmd .= "drop tablespace $tsname including
contents;\n";
            $cmd .= "create tablespace $tsname\n";
            $cmd .= "datafile
'$params{$tsname.'_area'}$ts_datafile' size
$params{$tsname.'_first_size'}
$params{$tsname.'_options'}\n";
            $cmd .= "extent management local\n";
            if (defined
($params{$tsname.'_extent_size'}) &&
```

```
($params{$tsname.'_extent_size'} =~
/[aA][uU][Tt][oO][Aa][lL][lL][oO][cC][aA][tT][eE]/))
            {
                $cmd .= "autoallocate\n";
            }
            elsif (defined
($params{$tsname.'_extent_size'}) &&
!($params{$tsname.'_extent_size'} =~
/[aA][uU][Tt][oO][Aa][lL][lL][oO][cC][aA][tT][eE]/))
            {
                $cmd .= "uniform size
".$params{$tsname.'_extent_size'}."\n";
            }
            else
            {
                $cmd .= "uniform";
            }
        }
    }
    $cmd .= " nologging\n" if
($params{$tsname.'_nolg'} =~ /[tT][rR][uU][eE]/);
    $cmd .= ";\n";
    &dump1("*sql");
}

sub add_ts_rollb
{
    $ts_undo_l = $_[0];
    @ts_datafiles = split(/,/,
$params{'ts_'.$ts_undo_l.'_datafiles'});
    $ts_datafile = shift (@ts_datafiles);
    $cmd .= "create tablespace ts_".$ts_undo_l."\n";
    $cmd .= "  datafile
'$params{'ts_undo_area'}$ts_datafile' size
$params{'ts_'.$ts_undo_l.'_first_size'}
$params{'ts_'.$ts_undo_l.'_options'};\n";
    &dump1("*sql");
    &dump0("*wait");
    &add_dfs("ts_$ts_undo_l");

# currently set up for the foreground - maybe should
be changed
    &dump0("# creating extra rollback segments");
    for ($i = 1; $i <=
$params{'ts_'.$ts_undo_l.'_#rs'}; $i++)
    {
        $cmd .= "create $params{'ts_undo_rs_type'}
rollback segment
$params{'ts_'.$ts_undo_l.'_rs_prefix'}$i";
        $cmd .= " storage
$params{'ts_undo_rs_storage'}" if defined
$params{'ts_'.$ts_undo_l.'_rs_storage'};
        $cmd .= " tablespace ts_$ts_undo_l" if
($params{'skip_ts'} !~ /$ts_undo_l/);
        $cmd .= ";\n";
    }
    &dump1("*sql");
    &dump0("*wait");

}

sub add_dfs
{
    $tsname = shift(@_);
    &dump0("# adding $params{'user'}'s $tsname
datafiles");
    @ts_datafiles = split(/,/,
$params{$tsname.'_datafiles'});
    $ts_datafile = shift (@ts_datafiles); # drop first
one
    $cur_inst = 2 if $multi;  # for possible locality
on SP2
    for ($i = 1; $i < $params{$tsname.'_#files'};
$i++)
    {
        $ts_datafile = shift (@ts_datafiles);
        &advmulti();
        $cmd .= "alter tablespace $tsname\n";
        if ($params{$tsname.'_managed'} =~
/[uU][sS][eE][rR]/)
        {  # user managed temporary tablespace
            $cmd .= "  add datafile ";
        }
        else
        {  # system managed temporary tablespace
            if ($params{$tsname.'_temporary'} =~
/true/)
            {
                $cmd .= "  add tempfile ";
            }
            else
            {
                $cmd .= "  add datafile ";
            }
        }
        $cmd .=
"'$params{$tsname.'_area'}$ts_datafile' size
$params{$tsname.'_size'}
$params{$tsname.'_options'};\n";
        &dump1("*sql");
    }
}

sub createios
{
    push (@iokv, "db_name=$params{'oracle_sid'}");
    push (@iokv,
"control_files=$params{'io_control_files'}");
    push (@iokv, "db_block_buffers=1000");
    push (@iokv, "shared_pool_size=35000000");
    push (@iokv, "parallel_max_servers=144");
    push (@iokv, "parallel_min_servers=0");
    push (@iokv, "max_dump_file_size=5000");
    push (@iokv, "audit_trail=FALSE");
    push (@iokv, "global_names=FALSE");
#    push (@iokv, "commit_point_strength=1");
#    push (@iokv, "dblink_encrypt_login=true");
#    push (@iokv, "db_block_size=16384");
    push (@iokv, "db_block_size=8192");
    push (@iokv, "db_file_multiblock_read_count=32");
    push (@iokv, "processes=256");
    push (@iokv, "sessions=256");
    push (@iokv, "transactions=512");
    push (@iokv,
"transactions_per_rollback_segment=20");
    push (@iokv, "max_rollback_segments=256");
    push (@iokv, "distributed_transactions=0");
    push (@iokv, "nls_date_format=YYYY-MM-DD");
    push (@iokv, "db_files=1023");
    push (@iokv, "open_cursors=1024");
    push (@iokv, "optimizer_mode=CHOOSE");
    push (@iokv, "optimizer_percent_parallel=100");
    push (@iokv, "sort_area_size=3000000");
    push(@iokv, "always_semi_join=HASH");
    push(@iokv, "parallel_broadcast_enabled=TRUE");
    push(@iokv, "optimizer_features_enable=8.0.4");
    push(@iokv, "compatible=8.0.4");
    push(@iokv, "hash_multiblock_io_count=64");
    push(@iokv, "always_anti_join=HASH");


    $ifile = sprintf ("%s%s", $params{'dbs_area'},
"init_$params{'oracle_sid'}.ora");
    $ifile =~ s/\?/$ENV{'ORACLE_HOME'}/g;
    if ($params{'skip_mk_initoras'} !~ /include/)
    {
        open (IOFILE, ">$ifile");
        print IOFILE "# Init.ora include file created
by bumpx.pl\n\n";
        while ($kv = shift(@iokv))
        {
            $kv =~ /(.*)=(.*)/;
            $entry = sprintf ("%-25s = %s\n", $1, $2);
            print IOFILE $entry;
        }
        if (defined $params{'io_ifile'})
        {
            $entry = sprintf ("%-25s = %s\n", "ifile",
$params{'io_ifile'});
            print IOFILE $entry;
        }
        close (IOFILE);
    }

    for ($j = 1; $j <= $num_inst; $j++)
    {
        $iofname[$j] = sprintf ("%s%s",
$params{'dbs_area'},
"init".$j."_$params{'oracle_sid'}.ora");
        $iofname_opn = $iofname[$j];
        $iofname_opn =~ s/\?/$ENV{'ORACLE_HOME'}/g;
        if ($params{'skip_mk_initoras'} !~ /instance/)
        {
            open (IOFILE, ">$iofname_opn");
            print IOFILE "# Init.ora for instance $j
created by bumpx.pl\n\n";
            print IOFILE "thread = $j\n"
                if ($params{'special_machine'} ne
'ncube');
            print IOFILE "ifile = $ifile\n";
```

```
            close (IOFILE);
        }
        $iofname[$j] = sprintf ("/host%s",
$iofname[$j])
            if ($params{'special_machine'} eq
'ncube');
    }
}


sub create_objects

{
    $ob_type = shift(@_);
    @ob_objectlist = @_;
    $cmd .= "connect
$params{'user'}/$params{'passwd'};\n";
    foreach $object (@ob_objectlist)
    {
        &create_object;
    }
    &dump1("*sql");
}


sub create_object

# this routine only creates one object
# need it because of the creation of materialized
views

{

    $ob_entry = $ob_type.'_'.$object;
    if (($ob_type cmp 'tab') == 0 )  # we are
creating a table
    {
        $cmd .= "drop table $object;\n";
        $cmd .= "create table $object";
    }
    elsif (($ob_type cmp 'view') == 0) # we are
creating a view
    {
        if ($params{$ob_entry.'_creation_type'} =~
/[dD][iI][rR][eE][cC][tT]/) # direct
        {
            $cmd .= "drop materialized view
$object;\n";
            $cmd .= "create materialized view
$object";
        }
        else           # with table
        {
            $cmd .= "drop table $object;\n";
            $cmd .= "create table $object";
        }
    }
    if (($ob_type cmp 'viewlog') == 0)       # we have
a viewlog
    {
        $cmd .= "drop materialized view log on
$object;\n";
        $cmd .= "create materialized view log ";
        $cmd .= "on $object\n";
    }
    elsif (!($ob_type eq 'view') ||
           (($ob_type eq 'view') &&
($params{$ob_entry.'_creation_type'} =~
/[wW][iI][tT][hH]\s[tT][aA][bB][lL][eE]/)))
    {
        $cmd .= "(\n";
        $cmd .= "$params{$ob_entry.'_cons'}\n" if
defined $params{$ob_entry.'_cons'};
        @ob_collist = split(/,/,
$params{$ob_entry.'_columns'});
        &add_columns;  # the list of columns are in
@ob_collist
        $cmd .= ")\n";
    }
    else
    {
        $cmd .= "\n";
    }
    if (defined $params{$ob_entry.'_cluster'})
    {
        $cmd .= "cluster $params{$ob_entry.'_cluster'}
($params{$ob_entry.'_clucols'})\n";
    }
    else
    {
        # add organization and pctthreshold for index
only tables
        $cmd .= "organization
$params{$ob_entry.'_org'}\n" if defined
$params{$ob_entry.'_org'};
        $cmd .= "pctthreshold
$params{$ob_entry.'_%t'}\n" if defined
$params{$ob_entry.'_%t'};
        $cmd .= "pctfree $params{$ob_entry.'_%f'}\n"
if defined $params{$ob_entry.'_%f'};
        $cmd .= "pctused $params{$ob_entry.'_%u'}\n"
if defined $params{$ob_entry.'_%u'};
        $cmd .= "initrans $params{$ob_entry.'_it'}\n"
if defined $params{$ob_entry.'_it'};
        $cmd .= "maxtrans $params{$ob_entry.'_mt'}\n"
if defined $params{$ob_entry.'_mt'};
        $cmd .= "tablespace
$params{$ob_entry.'_ts'}\n" if defined
$params{$ob_entry.'_ts'};
        $cmd .= "storage
$params{$ob_entry.'_storage'}\n" if defined
$params{$ob_entry.'_storage'};
        if (defined $params{$ob_entry.'_of_ts'})
        {
            # overflow tablespace specs
            $cmd .= "overflow ";
            $cmd .= "pctfree
$params{$ob_entry.'_of_%f'} " if defined
$params{$ob_entry.'_of_%f'};
            $cmd .= "pctused
$params{$ob_entry.'_of_%u'} " if defined
$params{$ob_entry.'_of_%u'};
            $cmd .= "initrans
$params{$ob_entry.'_of_it'} " if defined
$params{$ob_entry.'_of_it'};
            $cmd .= "maxtrans
$params{$ob_entry.'_of_mt'} " if defined
$params{$ob_entry.'_of_mt'};
            $cmd .= "tablespace
$params{$ob_entry.'_of_ts'} " if defined
$params{$ob_entry.'_of_ts'};
            $cmd .= "storage
$params{$ob_entry.'_of_storage'}\n" if defined
$params{$ob_entry.'_of_storage'};
        }
    }
    if ((defined $params{$ob_entry.'_pardeg'}) ||
(defined $params{$ob_entry.'_parinst'}))
    {
        $cmd .= "parallel";
        if ((defined $params{$ob_entry.'_pardeg'}) &&
            ($params{$ob_entry.'_pardeg'} =~
/[dD][eE][fF][aA][uU][lL][tT]/))
        {
            $cmd .= "\n";
        }
        else
        {
            $cmd .= "(degree
$params{$ob_entry.'_pardeg'} " if defined
$params{$ob_entry.'_pardeg'};
            $cmd .= "instances
$params{$ob_entry.'_parinst'}" if defined
$params{$ob_entry.'_parinst'};
            $cmd .= ")\n";
        }
    }
    $cmd .= "nologging\n" if
($params{$ob_entry.'_nolg'} =~ /[tT][rR][uU][eE]/);
    $cmd .= "cache\n" if ($params{$ob_entry.'_cache'}
=~ /[tT][rR][uU][eE]/);

    # add partition support
    if (defined $params{$ob_entry.'_#part'} &&
($params{$ob_entry.'_#part'} > 1))
    {
        &expand_partitions
($ob_entry,$object,$ob_type);
    }                  # no partitioning


    if (($ob_type cmp 'view') == 0)
    {
        $cmd .= "enable row movement\n" if
defined($params{$ob_entry.'_parttype'});
        if ($params{$ob_entry.'_creation_type'} =~
/[dD][iI][rR][eE][cC][tT]/)
        {
            $cmd .= "BUILD
".$params{$ob_entry.'_build_when'}."\n" if (defined
```

```
$params{$ob_entry.'_build_when'});
            &add_refresh_clause($ob_entry);
        }

        if ($params{$ob_entry.'_rewrite'} =~
/[tT][rR][uU][eE]/)
        {
            $cmd .= "enable query rewrite\n";
        }
        elsif ($params{$ob_entry.'_rewrite'} =~
/[fF][aA][lL][sS][eE]/)
        {
            $cmd .= "disable query rewrite\n";
        }
    }

    if (defined $params{$ob_entry.'_as_select'}) # as
select (here I only copy a variable called
<viewname>_define_as_select
    {
        &print_as_select_stm($params{$ob_entry.'_as_se
lect'});
        $cmd .= "\n";
    }

    if (($ob_type cmp 'viewlog') == 0)
    {
        $cmd .= "with rowid\n";
        if (defined $params{$ob_entry.'_columns'})
        {
            $cmd .= "(\n";
            @ob_collist = split(/,/,
$params{$ob_entry.'_columns'});
            &add_columns;      # the list of columns
are in @ob_collist
            $cmd .= ")\n";
        }
        $cmd .= "including new values\n";
    }

    if ($params{$ob_entry.'_creation_type'} =~
/[wW][iI][tT][hH]\s[tT][aA][bB][lL][eE]/)
    {
        $cmd .= ";\ndrop materialized view
$object;\n";
        $cmd .= "create materialized view $object \n";
        $cmd .= "BUILD
".$params{$ob_entry.'_build_when'}."\n" if (defined
$params{$ob_entry.'_build_when'});
        $cmd .= "on prebuilt table \n";
#       if (defined $params{$ob_entry.'_precision'})
#       {
#           ($params{$ob_entry.'_precision'} =~
/[rR][eE][dD][uU][cC][eE][dD]/) ?
#              $cmd .= "WITH REDUCED PRECISION \n"
#            : $cmd .= "WITHOUT REDUCED PRECISION
\n";
#       }

        &add_refresh_clause($ob_entry);

        if ((defined $params{$ob_entry.'_rewrite'}) &&
($params{$ob_entry.'_rewrite'} =~ /[tT][rR][uU][eE]/))
        {
            $cmd .= "enable query rewrite\n";
        }
        elsif ($params{$ob_entry.'_rewrite'} =~
/[fF][aA][lL][sS][eE]/)
        {
            $cmd .= "disable query rewrite\n";
        }

        $cmd .= "refresh " .
$params{$ob_entry.'_refresh'} ."\n" if (defined
$params{$ob_entry.'_refresh'});
        $cmd .= $params{$ob_entry.'_queryrw'} ."\n" if
(defined $params{$ob_entry.'_queryrw'});
        &print_as_select_stm($params{$ob_entry.'_as_se
lect'});
    }
    $cmd .= ";\n";
}

sub add_columns
{
    while ($col = shift(@ob_collist))
    {
        (@ob_collist == 0) ? ($addcomma = "") :
($addcomma = ",");
        if (($ob_type cmp 'tab') == 0 )
        {
            $columntype =
$params{$ob_entry.'_'.$col.'_type'};
        }
        else
        {
            $columntype = "";
        }
        $nextline = sprintf ("    %-20s %s
%s$addcomma\n", $col, $columntype,
$params{$ob_entry.'_'.$col.'_cons'});
        $cmd .= $nextline;
    }
}


sub add_refresh_clause

{
    $_ob_entry = $_[0];
    if (defined $params{$_ob_entry.'_refresh_how'})
    {
        $cmd .= "REFRESH
".$params{$_ob_entry.'_refresh_how'}."\n";
        $cmd .= "ON
".$params{$_ob_entry.'_refresh_when'}."\n" if (defined
$params{$_ob_entry.'_refresh_when'});
    }
}

sub print_as_select_stm
{
    if ($_[0] =~ /\/\*/)
    {
        ($beforehint,$rest)=split(/\/\*/,$_[0]);
        ($hint,$as_select) =split(/\*\//,$rest);
    }
    else
    {
        $as_select = $_[0];
        $hint="^";
    }
    $as_select =~ s/(, )|(,\t)/,\n /g;
    $as_select =~ s/from|FROM/\nFROM\n/g;
    $as_select =~ s/select|SELECT/SELECT\n/g;
    $as_select =~ s/where|WHERE/\nWHERE\n/g;
    $as_select =~ s/group by|GROUP BY/\nGROUP BY\n/g;
    $as_select =~ s/ and | AND /\n AND \n/g;
    $as_select =~ s/ or | OR /\n OR \n/g;
    $cmd .= "as select\n";
    $cmd .= "/*".$hint."*/"."\n" if ($hint !~ /\^/);
    $cmd .= $as_select;
}
sub create_clusters
{
    $cmd .= "connect
$params{'user'}/$params{'passwd'};\n";
    foreach $cluster (@clulist)
    {
        $cmd .= "drop cluster $cluster including
tables;\n";
        $cmd .= "create cluster $cluster (\n";
        @clu_collist = split(/,/,
$params{'clu_'.$cluster.'_columns'});
        while ($col = shift(@clu_collist))
        {
            (@clu_collist == 0) ? ($addcomma = "") :
($addcomma = ",");
            $nextline = sprintf ("    %-20s
%s$addcomma\n", $col,
$params{'clu_'.$cluster.'_'.$col.'_type'});
            $cmd .= $nextline;
        }
        $cmd .= ")\n";
        $cmd .= "pctfree
$params{'clu_'.$cluster.'_%f'}\n" if defined
$params{'clu_'.$cluster.'_%f'};
        $cmd .= "pctused
$params{'clu_'.$cluster.'_%u'}\n" if defined
$params{'clu_'.$cluster.'_%u'};
        $cmd .= "initrans
$params{'clu_'.$cluster.'_it'}\n" if defined
$params{'clu_'.$cluster.'_it'};
        $cmd .= "maxtrans
$params{'clu_'.$cluster.'_mt'}\n" if defined
$params{'clu_'.$cluster.'_mt'};
        $cmd .= "size
$params{'clu_'.$cluster.'_size'}\n" if defined
$params{'clu_'.$cluster.'_size'};
        $cmd .= "tablespace
```

```
$params{'clu_'.$cluster.'_ts'}\n" if defined
$params{'clu_'.$cluster.'_ts'};
        $cmd .= "storage
$params{'clu_'.$cluster.'_storage'}\n" if defined
$params{'clu_'.$cluster.'_storage'};
        $cmd .= "index\n" if
($params{'clu_'.$cluster.'_index'} =~ true);
        if (defined
$params{'clu_'.$cluster.'_hashkeys'})
        {
            $cmd .= "hash is
$params{'clu_'.$cluster.'_hashcol'} " if defined
$params{'clu_'.$cluster.'_hashcol'};
            $cmd .= "hashkeys
$params{'clu_'.$cluster.'_hashkeys'}\n";
        }
        if ((defined
$params{'clu_'.$cluster.'_pardeg'}) || (defined
$params{'clu_'.$cluster.'_parinst'}))
        {
            $cmd .= "parallel ";
            $cmd .= "degree
$params{'clu_'.$cluster.'_pardeg'} " if defined
$params{'clu_'.$cluster.'_pardeg'};
            $cmd .= "instances
$params{'clu_'.$cluster.'_parinst'}" if defined
$params{'clu_'.$cluster.'_parinst'};
            $cmd .= "\n";
        }
        $cmd .= "cache\n" if
$params{'clu_'.$cluster.'_cache'} =~ /true/;
        $cmd .= ";\n";
# Now, create the cluster index, if necessary
        if ($params{'clu_'.$cluster.'_index'} =~ true)
        {
            $cluindex = "ind_".$cluster;
            $cmd .= "drop index $cluindex;\n";
            $cmd .= "create index $cluindex on cluster
$cluster;\n";
        }
    }
    &dump1("*sql");
}


sub expand_partitions

# *MP* parameterize procedure: changed $tab_entry into
$ob_entry 1. parameter $_[0]
#                                       $table into $ob
2. parameter $_[1]
#                             introduced $ob_type
3. parameter $_[2]

{
    $ob_entry = $_[0];
    $ob = $_[1];
    $ob_type = $_[2];


    if ($params{$ob_entry.'_parttype'} =~
/[rR][aA][nN][gG][eE]/)
    {
        $cmd .= "partition by range (" .
$params{$ob_entry.'_partcol'} . ")\n(\n";
        &expand_partitions_doit($ob_entry,$ob,$ob_type
);
    }
    elsif ($params{$ob_entry.'_parttype'} =~
/[hH][aA][sS][hH]/)
    {
        $cmd .= "partition by hash (" .
$params{$ob_entry.'_partcol'} . ")\n";
        $cmd .= "partitions ".
$params{$ob_entry.'_#part'} . "\n";
    }
    elsif ($params{$ob_entry.'_parttype'} =~
/[cC][oO][mM][pP][oO][sS][iI][tT][eE]/)
    {
        $cmd .= "partition by range (" .
$params{$ob_entry.'_partcol'} . ")\n";
        $cmd .= "subpartition by hash(".
$params{$ob_entry.'_subpartcol'} . ")\n";
        $cmd .= "subpartitions ".
$params{$ob_entry.'_#subpart'} . "\n(\n";
        &expand_partitions_doit($ob_entry,$ob,$ob_type
);
    }
}                    # no partitioning


sub expand_partitions_doit

# *MP* parameterize procedure: changed $tab_entry into
$ob_entry 1. parameter $_[0]
#                                       $table into $ob
2. parameter $_[1]
#                             introduced $ob_type
3. parameter $_[2]

{
    $ob_entry = $_[0];
    $ob = $_[1];
    $ob_type = $_[2];

    @pcollist = split(/,/,
$params{$ob_entry.'_partcol'});
    if (!defined $params{$ob_entry.'_partnames'})
    {
        # if no partnames are specified, use a default
part name
        for ($i = 1; $i <=
$params{$ob_entry.'_#part'}; $i++)
        {
            ($i == $params{$ob_entry.'_#part'}) ?
($addcomma = "") :
            ($addcomma = ",");
            $nextfile = sprintf("%s%s%d%s", $ob,"_p",
$i, $addcomma );
            $params{$ob_entry.'_partnames'} =
                $params{$ob_entry.'_partnames'} .
$nextfile;
        }
    }

    @ob_partnames =
split(/,/,$params{$ob_entry.'_partnames'});

    # if partnames is specified as XXXX#, then expand
    if ($ob_partnames[0] =~ /\#/)
    {
        $filenm = shift(@ob_partnames);
        $savename = $filenm;
        $params{$ob_entry.'_partnames'} = "";
        for ($i = 1; $i <=
$params{$ob_entry.'_#part'}; $i++)
        {
            $filenm =~ s/\#/$i/g;
            ($i == $params{$ob_entry.'_#part'}) ?
($addcomma = "") :
            ($addcomma = ",");
            $params{$ob_entry.'_partnames'} =
                $params{$ob_entry.'_partnames'} .
$filenm . $addcomma;
            $filenm = $savename;
        }
        @ob_partnames =
split(/,/,$params{$ob_entry.'_partnames'});
        printf("Expanded $savename
to:\n$params{$ob_entry.'_partnames'}\n\n") if
$verbose;
    } else {
        if (@ob_partnames !=
$params{$ob_entry.'_#part'})
        {
            print "Number of partitions
$params{$ob_entry.'_#part'} for $ob doesn't match\n
_partnames parameter for
$params{$ob_entry.'_partnames'}\n";
            exit(-1);
        }
    }
    # now process the partition boundary

    &process_part_brys ($ob_entry,$ob);
    &process_part_ts ($ob_entry,$ob);

    # complete the partition statement

    for ($i=0; $i < $params{$ob_entry.'_#part'}; $i++)
    {
        $cmd .= "partition " . $ob_partnames[$i] . "
values less than ";
        if ($i==$params{$ob_entry.'_#part'}-1) {
          $cmd .= "(MAXVALUE)\n";
        }
        else {
          $cmd .= "(" . $ob_part_vals[$i] . ")\n";
        }
        &process_part_params($ob_type,'%f','pctfree',$
```

```
ob,$ob_partnames[$i]);
        &process_part_params($ob_type,'it','initrans',
$ob,$ob_partnames[$i]);
        &process_part_params($ob_type,'mt','maxtrans',
$ob,$ob_partnames[$i]);

        if (defined
$params{$ob_entry.'_'.$ob_partnames[$i].'_ts'})
        {
            $cmd .= 'tablespace ' .
$params{$ob_entry.'_'.$ob_partnames[$i].'_ts'} . "\n";

&process_part_storage($ob_type,$ob,$ob_partnames[$i]);
        } elsif (defined
$params{$ob_entry.'_part_ts'}) {
            $cmd .= 'tablespace ' . $ob_part_ts[$i] .
"\n";

&process_part_storage($ob_type,$ob,$ob_partnames[$i]);
        }
        if (defined
$params{$ob_entry.'_'.$ob_partnames[$i].'_nolg'})
        {
            $cmd .= "nologging\n" if
($params{$ob_entry.'_'.$ob_partnames[$i].'_nolg'} =~
/[tT][rR][uU][eE]/);
        } elsif (defined
$params{$ob_entry.'_part_def_nolg'}) {
            $cmd .= "nologging\n" if
($params{$ob_entry.'_part_def_nolg'} =~
/[tT][rR][uU][eE]/);
        }
        $cmd .= (($i+1) ==
$params{$ob_entry.'_#part'}) ? "" : ",\n";
    }
    $cmd .= ")\n";
} #end expand_partitions_doit


sub process_part_storage
{
    local($type,$tname,$pname) = @_;
    if (defined
$params{$type.'_'.$tname.'_'.$pname.'_storage'})
    {
        $cmd .= 'storage
'.$params{$type.'_'.$tname.'_'.$pname.'_storage'}."\n"
;
    } elsif (defined
$params{$type.'_'.$tname.'_part_def_storage'}) {
        $cmd .= 'storage
'.$params{$type.'_'.$tname.'_part_def_storage'}."\n";
    }
}


sub process_part_params
{
    local($type,$p1,$p2,$tname,$pname) = @_;
    if (defined
$params{$type.'_'.$tname.'_'.$pname.'_'.$p1}) {
        $cmd .= $p2 . " " .
$params{$type.'_'.$tname.'_'.$pname.'_'.$p1}."\n";
    } elsif (defined
$params{$type.'_'.$tname.'_part_def_'.$p1}) {
        $cmd .= $p2 . " " .
$params{$type.'_'.$tname.'_part_def_'.$p1} . "\n";
    }
}


sub process_part_ts

# *MP* parameterize procedure: changed 'tab'.$table
into $ob__entry 1. parameter $_[0] (same as $_entry in
expand_partition)
#                                       $table into
$ob_             2. parameter $_[1] (same as $ob in
expand_partition)

{
    $ob__entry = $_[0];
    $ob_ = $_[1];
    # is objecttype_<name>_part_ts is in the form of
XXXX#, expand it
    # else treat it as a comma separated list of ts
names
    if (defined $params{$ob__entry.'_part_ts'})
    {
        $nts=$params{$ob__entry.'_part_ts'};
        $nts=~s/#//;
        $nts=$params{$nts."_instances"};
        @ob_part_ts =
split(/,/,$params{$ob__entry.'_part_ts'});
        if ($ob_part_ts[0] =~ /\#/)
        {
            $filenm = shift(@ob_part_ts);
            $savename = $filenm;
            $params{$ob__entry.'_part_ts'} = "";
            $its = 0;
            for ($i = 1; $i <=
$params{$ob__entry.'_#part'}; $i++)
            {
                $its = $its + 1;
                if ($its > $nts) {
                    $its=1;
                }
                $filenm =~ s/\#/$its/g;
                ($i == $params{$ob__entry.'_#part'}) ?
($addcomma = "") :
                    ($addcomma = ",");
                $params{$ob__entry.'_part_ts'} =
                    $params{$ob__entry.'_part_ts'} .
$filenm . $addcomma;
                $filenm = $savename;
            }
            @ob_part_ts =
split(/,/,$params{$ob__entry.'_part_ts'});
        } else {
            if (@ob_part_ts !=
$params{$ob__entry.'_#part'})
            {
                if (($params{$ob__entry.'_#part'} %
@ob_part_ts) == 0) {
                    $numfil = @ob_part_ts;
                    $p = $params{$ob__entry.'_#part'} /
$numfil;
                    $fil = "";
                    for ($i = 0; $i < $p; $i++)
                    {
                        if ($i == 0) {
                            $fil =
$params{$ob__entry.'_part_ts'};
                        } else {
                            $fil =
join(',',$fil,$params{$ob__entry.'_part_ts'});
                        }
                    }
                    @ob_part_ts = split(/,/,$fil);
                } else {
                    print "Number of partitions
$params{$ob__entry.'_#part'} for object $ob_\n doesn't
match $table_part_ts parameter.\n";
                    exit (-1);
                }
            }
        }
    }
} # end of process_part_ts


sub process_part_brys


# *MP* parameterize procedure: changed 'tab'.$table
into $ob__entry 1. parameter $_[0] (same as $ob_entry
in expand_partition)
#                                       $table into
$ob_             2. parameter $_[1] (same as $ob in
expand_partition)

{
    $ob__entry = $_[0];
    $ob_ = $_[1];

    $cnt = 0;
    @ob_part_vals = ();

    foreach $col (@pcollist)
    {
        # add quotes for character strings and dates


        if (($params{$ob__entry.'_'.$col.'_type'} =~
/[cC][hH][aA][rR]/) ||
            ($params{$ob__entry.'_'.$col.'_type'} =~
/[dD][aA][tT][eE]/))
        {
            $addquote = "'";
        } else {
```

```
            $addquote = "";
        }

        @ob_part_col =
split(/,/,$params{$ob__entry.'_'.$col.'_partvals'});
        if (@ob_part_col !=
$params{$ob__entry.'_#part'})
        {
            printf "Number of partition boundary
values %d for column $col in object $ob_ doesn't match
the number of partitions
($params{$ob__entry.'_#part'}) of the object\n",
($#ob_part_col + 1);
            exit(-1);
        }
        for ($i=0; $i < $params{$ob__entry.'_#part'};
$i++)
        {
            ($cnt == $#pcollist) ? ($addcomma = "") :
($addcomma = ",");
            if ($ob_part_col[$i] =~
/[Mm][Aa][Xx][Vv][Aa][Ll][Uu][Ee]/) {
                $addquote = "";
            }
            if ($params{$ob__entry.'_'.$col.'_type'}
=~ /[dD][aA][tT][eE]/)
            {
                if ($i == $params{$ob_entry.'_#part'}
- 1)
                {
                   $ob_part_vals[$i] .= "MAXVALUE";
                } else {
                   $nls_format = (defined
$params{$ob__entry.'_'.$col.'_date_format'}) ?
$params{$ob__entry.'_'.$col.'_date_format'} : "YYYY-
MM-DD";
                    $ob_part_vals[$i] .=
"to_date('".$ob_part_col[$i]."', '";
                    $ob_part_vals[$i] .=
$nls_format."')".$addcomma;
                }
            } else {
                $ob_part_vals[$i] = $ob_part_vals[$i] .
$addquote.
                    $ob_part_col[$i] . $addquote .
$addcomma ;
            }
        }
        $cnt++;
    }
} # end of process_part_brys

sub alttabs
{
    $di = shift(@_);
    $di =~ /(\d*)_(\d*)/;
    $cmd = "";
    $cmd .= "connect
$params{'user'}/$params{'passwd'};\n";
    $params{'alter_tables'} =  $params{'tab_tables'}
if !defined $params{'alter_tables'};
    @atabs = split(/,/, $params{'alter_tables'});
    foreach $atab (@atabs)
    {
        $cmd .= "alter table $atab parallel (degree $1
instances $2);\n";
    }
    &dump1("*sql");
    &dump0("*wait");
}

sub setinst
{
    $inum = shift(@_);
    &dump0("*inst");
    &dump0("{$inum}");
}

sub dump2
{
    $value = shift(@_);
    &dump0($value);
    &dump0("{");
    print OUTFILE $cmd;
    &dump0("}");
    &dump0("{");
    print OUTFILE $cmd2;
    &dump0("}");
    $cmd = "";
    $cmd2 = "";
    print "." if $verbose;
}

sub dump1
{
    $value = shift(@_);
    &dump0($value);
    &dump0("{");
    print OUTFILE $cmd;
    &dump0("}");
    $cmd = "";
    print "." if $verbose;
}

sub dump0
{
    $value = shift (@_);
    $value = $value . "\n" if ($value !~ /.*\n$/);
    print OUTFILE "$value";
}

sub recreate_drive_extended_part
{
    $cmd = "creapart -d PhysicalDrive" . $drivenum .
"\n";
    &dump1("*sh");
    $cmd = "Deleted partitions on PhysicalDrive" .
$drivenum;
    &time0($cmd);
    $cmd = "creapart -x PhysicalDrive" . $drivenum .
"\n";
    &dump1("*sh");
    $cmd = "Created extended partition on
PhysicalDrive" . $drivenum;
    &time0($cmd);
}

sub create_drive_part
{
    $numfiles = $params{$ts_entry.'_#files'}-1;
    if ($d == 0)
    {
        $size = (defined
$params{$ts_entry.'_first_size'}) ?
                $params{$ts_entry.'_first_size'} :
                $params{$ts_entry.'_size'};
    }
    else
    {
        $size = $params{$ts_entry.'_size'};
    }
    $size =~ s/[Mm]*//g;
    @files =
split(/,/,$params{$ts_entry.'_datafiles'});
    foreach $file (@files)
    {
        &create_drive_part_file;
    }

}

sub create_drive_part_file
{
#   add 1MB to nt partition size to prevent writing
data to cyl 0
    $psize = $size+1;
    $drivenum = $file =~ /^log/ ?
$params{'plcre_log_drivenum'} :
        @drivenum[$d];
    $cmd = "creapart -l PhysicalDrive";
    $cmd .= $drivenum . " " . $psize . "\n";
    &dump1("*sh");
    $cmd = $file . "\tdevice\\PhysicalDrive" .
$drivenum;
    $cmd .=    "\\partition" . ++$partnum{$drivenum} .
"\n";
    &time0($cmd);
    print LNKSFILE $cmd;
    if ($file =~ /^sys/ || $file =~ /^cntr/)
    {
        $d = $d < $md ? ++$d : 0;
    }
    elsif ($file =~ !/^log/)
    {
        $d = ($d < $md) && ($d < $numfiles) ? ++$d :
0;
    }
}

sub load_ctl_head
```

```
{
    print "control $control\n";
    open (CTLFILE, ">$control");
    print CTLFILE "---\n";
    print CTLFILE "---  $table.ctl for delimited
records\n" if ($ud == 1);
    print CTLFILE "---  $table.ctl for fixed-length
fields\n" if ($ud == 0);
    print CTLFILE "---\n\n";
    if (!($pre72) && defined
$params{'tab_'.$table.'_loadextent'})
    {
        print CTLFILE "options\n";
        print CTLFILE "(\n";
        print CTLFILE "storage = (initial
$params{'tab_'.$table.'_loadextent'} next
$params{'tab_'.$table.'_loadextent'})\n";
        print CTLFILE ")\n";
    }
    print CTLFILE "unrecoverable\n" if
($params{'load_unrecoverable'} =~ /[tT][rR][uU][eE]/);
    print CTLFILE "load\n";
    print CTLFILE "-- This is where INFILE should
go.\n";
}

sub load_ctl_tail
{
    print CTLFILE "$params{'load_insert_type'}\n";
    print CTLFILE "fields terminated by
$params{'load_field_terminator'}\n" if ($ud == 1);
    print CTLFILE "(\n";
    @tab_collist = split(/,/,
$params{'tab_'.$table.'_columns'});
    while ($col = shift(@tab_collist))
    {
        (@tab_collist == 0) ? ($addcomma = "") :
($addcomma = ",");
        $pos = "";
        $pos = sprintf ("position
(%s)",$params{'tab_'.$table.'_'.$col.'_pos'}) if ($ud
== 0);
        $ctlline = sprintf ("  %-20s %s %s$addcomma",
$col, $pos,
$params{'tab_'.$table.'_'.$col.'_loadcolx'});
        print CTLFILE "$ctlline\n";
    }
    print CTLFILE ")\n";
    close (CTLFILE);
}

#*****************************************************
************************
#*****************************************************
************************
# Assigning Defaults
#*****************************************************
************************
#*****************************************************
************************

sub defaults
{
# defaults for the params associative array
    if (defined $bmtype)
    {
        eval(&$bmtype);
        &assigndefs;
    }
    &auxdefaults;
    &assigndefs;


$params{'dd_sql_area'}="$ENV{'ORACLE_HOME'}/rdbms/admi
n/" if !(defined $params{'dd_sql_area'});

$params{'dd_sqlplus_area'}="$ENV{'ORACLE_HOME'}/sqlplu
s/admin/" if !(defined $params{'dd_sqlplus_area'});

    if (defined $params{'compatible'})
    {
        $pre72 = 1 if $params{'compatible'} =~
/7\.(0|1)/;
        $pre73 = 1 if $params{'compatible'} =~
/7\.(0|1|2)/;
    }
    $params{'dbs_area'} =~ s/\?/$ENV{'ORACLE_HOME'}/g;
    $params{'load_tables'} = $params{'tab_tables'} if
!defined $params{'load_tables'};
    foreach $phase (@phases)
    {
        $params{$phase.'_num_inst'} =
$params{'def_num_inst'} if !defined
$params{$phase.'_num_inst'};
        $params{$phase.'_max_bg'} = $params{'max_bg'}
if !defined $params{$phase.'_max_bg'};
        $params{'startupfile_'.$phase} =
$params{'io_ifile'} if
!defined($params{'startupfile_'.$phase});
        $params{'startupfile_'.$phase} = sprintf
("/host%s", $params{'startupfile_'.$phase})
            if ($params{'special_machine'} eq
'ncube');
        #$params{'startupfile_'.$phase} =~
s/\?/$ENV{'ORACLE_HOME'}/g;
    }
    $params{'ops_nodes'} = 'undo' if
!defined($params{'ops_nodes'});
    @ops_nodes = split(/,/,$params{'ops_nodes'}) if
defined $params{'ops_nodes'};
    $params{'db_maxinstances'} = @ops_nodes if
!defined $params{'db_maxinstances'};

    #------------------------ undo tablespace
definition -----------------------------------
    $params{'ts_undo_#rs'} = $params{'def_num_inst'}
if !defined $params{'ts_undo_#rs'};
    $params{'ts_undo_area'} = $params{'dbs_area'} if
!defined $params{'ts_undo_area'};
    $params{'ts_undo_rs_prefix'} = 'r' if !defined
$params{'ts_undo_rs_prefix'};
    if (!defined $params{'ts_undo_rs_type'}){
#rs_type can be public or private
        if (@ops_nodes > 1) {
            $params{'ts_undo_rs_type'} = 'private';
        }else{
            $params{'ts_undo_rs_type'} = 'public';
        }
    }
    $save_datafiles = $params{'ts_undo_datafiles'};

&expand_filename('ts_undo_datafiles',$params{'ts_undo_
#files'});

    if (@ops_nodes > 1)
    {
        foreach $node (@ops_nodes)
        {
            $params{'ts_undo_'.$node.'_#files'} =
$params{'ts_undo_#files'}
            if
!defined($params{'ts_undo_'.$node.'_#files'});

            $df= $save_datafiles.'_'.$node;

            $params{'ts_undo_'.$node.'_datafiles'} =
$df
                if
!defined($params{'ts_undo_'.$node.'_datafiles'});

&expand_filename('ts_undo_'.$node.'_datafiles',$params
{'ts_undo_'.$node.'_#files'});

            $params{'ts_undo_'.$node.'_first_size'} =
$params{'ts_undo_first_size'}
            if
!defined($params{'ts_undo_'.$node.'_first_size'});

            $params{'ts_undo_'.$node.'_size'} =
$params{'ts_undo_size'}
                if
!defined($params{'ts_undo_'.$node.'size'});

            $params{'ts_undo_'.$node.'_#rs'} =
$params{'ts_undo_#rs'}
            if
!defined($params{'ts_undo_'.$node.'_#rs'});

            $params{'ts_undo_'.$node.'_rs_storage'} =
$params{'ts_undo_rs_storage'}
            if
!defined($params{'ts_undo_'.$node.'_rs_storage'});

            $pr = $params{'ts_undo_rs_prefix'}.$node;
            $params{'ts_undo_'.$node.'_rs_prefix'} =
$pr
                if
!defined($params{'ts_undo_'.$node.'_rs_prefix'});
        }
    }
```

```
    $params{'ts_log_#files'} =
$params{'ts_log_#threads'} *
$params{'ts_log_#files_pt'};
    $params{'ts_def_area'} = $params{'dbs_area'} if
!defined $params{'ts_def_area'};
    $params{'load_flatfile_area'} =
$params{'ts_def_area'} if !defined
$params{'load_flatfile_area'};
    $params{'load_controlfile_area'} =
$params{'ts_def_area'} if !defined
$params{'load_controlfile_area'};
    $params{'load_otherfile_area'} =
$params{'ts_def_area'} if !defined
$params{'load_otherfile_area'};

    $params{'ts_index_names'} = "ts_index" if
((!defined $params{'ts_index_names'}) &&
($params{'skip_ts'} !~ /index/));
    $params{'ts_temp_names'} = "ts_temp" if ((!defined
$params{'ts_temp_names'}) && ($params{'skip_ts'} !~
/temp/));
    $params{'ts_data_names'} = "ts_data" if ((!defined
$params{'ts_data_names'}) && ($params{'skip_ts'} !~
/data/));

    # expand ts_data groups

    if (defined $params{'ts_data_groups'})
    {
        @ts_data_group =
split(/,/,$params{'ts_data_groups'});
        foreach $gname (@ts_data_group)
        {
            $nts = $params{$gname.'_group_#ts'};
            for ($i=0;$i<$nts;$i++)
            {
                $tsn = sprintf("%s%d",$gname,($i+1));
                $params{'ts_data_names'} .= ",".$tsn;
                $params{$tsn.'_datafiles'} = $tsn."_#";
                $params{$tsn.'_#files'} =
$params{$gname.'_group_#files'} if defined
$params{$gname.'_group_#files'};
                $params{$tsn.'_option'} =
$params{$gname.'_group_option'} if defined
$params{$gname.'_group_option'};
                $params{$tsn.'_area'} =
$params{$gname.'_group_area'} if defined
$params{$gname.'_group_area'};
                $params{$tsn.'_storage'} =
$params{$gname.'_group_storage'} if defined
$params{$gname.'_group_storage'};
                $params{$tsn.'_first_size'} =
$params{$gname.'_group_first_size'} if defined
$params{$gname.'_group_first_size'};
                $params{$tsn.'_size'} =
$params{$gname.'_group_size'} if defined
$params{$gname.'_group_size'};
                $params{$tsn.'_nolg'} =
$params{$gname.'_nolg'} if defined
$params{$gname.'_nolg'};
            }
        }
    }

    @ts_data = split(/,/, $params{'ts_data_names'});
    @ts_index = split(/,/, $params{'ts_index_names'});
    @ts_temp = split(/,/, $params{'ts_temp_names'});

# expand tablespaces
    @ts_data_new =();
    $ts_data_names="@@";
    foreach $ts_entry (@ts_data)
    {
        if (defined($params{$ts_entry.'_instances'})){
            for ($i = $params{$ts_entry.'_instances'};
$i > 0; $i --) {
                $ts_new_entry = $ts_entry.$i;
                $params{$ts_new_entry.'_size'}=$params{
$ts_entry.'_size'};
                $params{$ts_new_entry.'_datafiles'}=$pa
rams{$ts_entry.'_datafiles'};
                $params{$ts_new_entry.'_datafiles'}=~s/
\#/$i/;
                $params{$ts_new_entry.'_#files'}=$param
s{$ts_entry.'_#files'};
                $params{$ts_new_entry.'_storage'}=$para
ms{$ts_entry.'_storage'};
                $params{$ts_new_entry.'_managed'}=$para
ms{$ts_entry.'_managed'}if
defined($params{$ts_entry.'_managed'});

$params{$ts_new_entry.'_extent_size'}=$params{$ts_entr
y.'_extent_size'}if
defined($params{$ts_entry.'_extent_size'});
                unshift (@ts_data_new, $ts_new_entry);
                $ts_data_names=$ts_data_names.",".$ts_n
ew_entry;
            }
        }
        else {
            unshift (@ts_data_new, $ts_entry);

$ts_data_names=$ts_data_names.",".$ts_entry;
        }
    }
    $ts_data_names=~s/@@,//;
    #print "ts_data_names: $ts_data_names\n";
    @ts_data = @ts_data_new;
    #print "ts_all before substitution @ts_all\n";
    #print "ts_all_new before substitution
@ts_all_new\n";
    #while (@ts_data) {
#       $ts = shift (@ts_data);
        #unshift (@ts_data_new,$ts);
    #}
    #foreach $ts_entry (@ts_data)
    #{
#       print "ts_entry: $ts_entry\n";
#       print "size: $params{$ts_entry.'_size'}\n";
#       print "datafiles:
$params{$ts_entry.'_datafiles'}\n";
#       print "#files:
$params{$ts_entry.'_#files'}\n";
#       print "storage:
$params{$ts_entry.'_storage'}\n";
#   }

    #print "ts_all after substitution @ts_all\n";
    #print "$ts_list\n";
    #exit(0);
    $mostts =
"$params{'ts_temp_names'},$ts_data_names,$params{'ts_i
ndex_names'}";
    $allts =
"ts_sys,ts_log,$params{'ts_undo'},$params{'ts_temp_nam
es'},$ts_data_names,$params{'ts_index_names'}";
    @ts_most = split(/,/, $mostts);
    @ts_all = split(/,/, $allts);

    foreach $ts_entry (@ts_most)
    {
        $params{$ts_entry.'_#files'} =
$params{'ts_def_#files'} if !defined
$params{$ts_entry.'_#files'};
        $params{$ts_entry.'_size'} =
$params{'ts_def_size'} if !defined
$params{$ts_entry.'_size'};
        $params{$ts_entry.'_storage'} =
$params{'ts_def_storage'} if !defined
$params{$ts_entry.'_storage'};
    }


    #rint "ts_all: @ts_all\n";
    foreach $ts_entry (@ts_all)
    {
        $params{$ts_entry.'_area'} =
$params{'ts_def_area'} if !defined
$params{$ts_entry.'_area'};
        $params{$ts_entry.'_options'} =
$params{'ts_def_options'} if !defined
$params{$ts_entry.'_options'};
        $params{$ts_entry.'_first_size'} =
$params{$ts_entry.'_size'} if !defined
$params{$ts_entry.'_first_size'};
        $params{$ts_entry.'_temporary'} = "false" if
!defined $params{$ts_entry.'_temporary'};
        $params{$ts_entry.'_managed'} = "user" if
!defined $params{$ts_entry.'_managed'};
        if (!defined $params{$ts_entry.'_datafiles'})
        {
            for ($i = 0; $i <
$params{$ts_entry.'_#files'}; $i++)
            {
                (($i+1) ==
$params{$ts_entry.'_#files'}) ? ($addcomma = "") :
($addcomma = ",");
                $nextfile = sprintf ("%s%d.f%s",
$ts_entry, $i+1, $addcomma);
```

```
                $params{$ts_entry.'_datafiles'} =
$params{$ts_entry.'_datafiles'} . $nextfile;
            }
        }
        @ts_datafiles = split(/,/,
$params{$ts_entry.'_datafiles'});
        $cntr = 1;
        if ($ts_datafiles[0] =~ /\#/) # we want to
replace all #'s
        {
            $filenm = shift(@ts_datafiles);
            $savename = $filenm;
            $params{$ts_entry.'_datafiles'} = "";
            for ($i = 0; $i <
$params{$ts_entry.'_#files'}; $i++)
            {
                $filenm =~ s/\#/$cntr/g;
                $cntr++;
                (($i+1) ==
$params{$ts_entry.'_#files'}) ? ($addcomma = "") :
($addcomma = ",");
                $params{$ts_entry.'_datafiles'} =
$params{$ts_entry.'_datafiles'} .
                    $filenm . $addcomma;
                $filenm = $savename;
            }
            printf ("Expanded $savename
to:\n$params{$ts_entry.'_datafiles'}\n\n") if
$verbose;
        }
        else
        {
            if (@ts_datafiles !=
$params{$ts_entry.'_#files'})
            {
                print "Number of files
($params{$ts_entry.'_#files'}) for $ts_entry doesn't
match\n _datafile parameter
$params{$ts_entry.'_datafiles'}\n";
                exit(-1);
            }
        }
    }
    $params{'io_control_files'} =
$params{'ts_sys_area'}.'t_cf1.f' if !defined
$params{'io_control_files'};

    @tab_list = split(/,/, $params{'tab_tables'});
    foreach $table (@tab_list)
    {
        $params{'load_dbgen_'.$table.'_option_C'} =
$params{'load_dbgen_def_option_C'} if !defined
$params{'load_dbgen_'.$table.'_option_C'};

        $params{'tab_'.$table.'_ts'} = "ts_data" if (
(!defined $params{'tab_'.$table.'_ts'}) && (!defined
$params{'tab_'.$table.'_part_ts'}));
        if (defined($params{'load_deg_parallel'})) {
            $params{'tab_'.$table.'_load_degpar'} =
$params{'load_deg_parallel'} if !defined
$params{'tab_'.$table.'_load_degpar'};
        } else {
            $params{'tab_'.$table.'_load_degpar'} = 1
if !defined $params{'tab_'.$table.'_load_degpar'};
        }

        #$params{'load_dbgen_'.$table.'_output_prefix'
} = "$table.tbl" if !defined
$params{'load_dbgen_'.$table.'_output_prefix'};

        $params{'tab_'.$table.'_load_datf'} =
"$table.tbl" if !defined
$params{'tab_'.$table.'_load_datf'};

        $params{'tab_'.$table.'_load_badf'} =
"$table.badf" if !defined
$params{'tab_'.$table.'_load_badf'};

        $params{'tab_'.$table.'_load_ctlf'} =
"$table.ctl" if !defined
$params{'tab_'.$table.'_load_ctlf'};

        $params{'tab_'.$table.'_load_logf'} =
"$table.log" if !defined
$params{'tab_'.$table.'_load_logf'};

        $params{'tab_'.$table.'_load_direct'} = "true"
if !defined $params{'tab_'.$table.'_load_direct'};

        $params{'tab_'.$table.'_#rows'} *=
$params{'scale_factor'};
    }

    @view_list = split(/,/, $params{'view_views'});
    foreach $view (@view_list)
    {
        $params{'view_'.$view.'_ts'} = "ts_data" if
!defined $params{'view_'.$view.'_ts'};
        $params{'view_'.$view.'_load_degpar'} =
$params{'load_deg_parallel'} if !defined
$params{'view_'.$view.'_load_degpar'};
        $params{'view_'.$view.'_#rows'} *=
$params{'scale_factor'};
        $params{$ob_entry.'_rewrite'} = "" if !defined
$params{$ob_entry.'_rewrite'};
    }

    @tablelog_list = split(/,/,
$params{'tablelog_list'});

    @clulist = split(/,/, $params{'clu_clusters'});
    foreach $cluster (@clulist)
    {
        $params{'clu_'.$cluster.'_ts'} = "ts_data" if
!defined $params{'clu_'.$cluster.'_ts'};
    }

    @constlist = split(/,/,
$params{'con_constraints'});
    foreach $const (@constlist)
    {
        $params{'con_'.$const.'_ts'} = "ts_index" if
((!defined $params{'con_'.$const.'_ts'}) &&
($params{'skip_ts'} !~ /index/));
    }

    @indexlist = split(/,/, $params{'ind_indices'});
    foreach $index (@indexlist)
    {
        $params{'ind_'.$index.'_ts'} = "ts_index" if
((!defined $params{'ind_'.$index.'_ts'}) &&
($params{'skip_ts'} !~ /index/));
    }
    $params{'analyze_type'} = "via package dbms.stats"
if !defined($params{'analyze_type'});
    @anlyzlist = split(/,/, $params{'anl_objects'});
    @anlyzlist = "schema" if (@anlyzlist == 0);
    foreach $object (@anlyzlist)
    {
        $params{'anl_'.$object.'_object_type'} =
"schema" if ($object =~ /schema/);
        $params{'anl_'.$object.'_object_type'} =
"table" if !defined
$params{'anl_'.$object.'_object_type'};
        $params{'anl_'.$object.'_anl_type'} = "gather"
if !defined $params{'anl_'.$object.'_anl_type'};
    }
    @choblist = split(/,/, $params{'chob_objects'});
#    foreach $object (@choblist)
#    {
#      $choblist{$object} =
$params{'chob_'.$object.'_pardeg'};
#    }
    @chstorlist = split(/,/,
$params{'chstor_objects'});
    foreach $object (@chstorlist)
    {
      $chstorlist{$object} =
$params{'chstor_'.$object.'_next'};
    }
    if ((defined $params{'verbose'}) &&
(($params{'verbose'} =~ /[tT][rR][uU][eE]/) ||
($params{'verbose'} =~ 1))){
        $verbose=1;}
    else {
        $verbose=0;}
    if ((defined $params{'log_output'}) &&
(($params{'log_output'} =~ /[tT][rR][uU][eE]/) ||
($params{'log_output'} =~ 1))){
        $log_output=1;}
    else {
        $log_output=0;}
    $params{'plcre_recreate_extended_partitions'} =
"false" if !defined
($params{'plcre_recreate_extended_partitions'});
}

sub assigndefs
{
    while ($onedef = shift(@defparams))
```

```
    {
        $onedef =~ /^(.*)=(.*)$/;
        $key = $1;
        $value = $2;
        $params{$key} = $value if !defined
$params{$key};
    }
}

sub auxdefaults
{
    @defparams = ();
    push (@defparams, 'scale_factor=0.1');
    push (@defparams, 'dbs_area=?/dbs/');
    push (@defparams, 'dbs_area=/tmp');
    push (@defparams, 'dd_sql_area=?/rdbms/admin/');
    push (@defparams, 'max_bg=-1');
    push (@defparams, 'user=bmuser');
    push (@defparams, 'passwd=bmpasswd');
    push (@defparams, 'load_type=delim');
    push (@defparams,
'load_field_terminator=whitespace');
    push (@defparams, 'ts_def_#files=1');
    push (@defparams, 'ts_def_options=reuse');
    push (@defparams, 'ts_def_first_size=1m');
    push (@defparams, 'ts_def_size=1m');
    push (@defparams, 'ts_sys_#files=1');
    push (@defparams, 'ts_log_#threads=1');
    push (@defparams, 'ts_log_#files_pt=2');
    push (@defparams, 'ts_undo_#files=1');
    push (@defparams, 'ts_sys_size=25m');
    push (@defparams, 'ts_log_size=2m');
    push (@defparams, 'ts_undo_size=10m');
#    push (@defparams, 'ts_data_names=ts_data');
    push (@defparams, 'load_insert_type=append');
    push (@defparams, 'load_deg_parallel=1');
    push (@defparams, 'special_machine=');
    push (@defparams, 'shutdown_level=abort');
    push (@defparams, 'def_num_inst=1');
#    push (@defparams, 'privileges=resource,unlimited
tablespace,connect');
    #push (@defparams, 'privileges=DBA,query
rewrite,global query rewrite');
    push (@defparams, 'privileges=DBA');
    push (@defparams, 'qry_number_trials=1');
}

sub tpcd
{
    @defparams = ();
    push (@defparams, 'user=tpcd');
    push (@defparams, 'passwd=tpcd');
    push (@defparams,
'tab_tables=lineitem,orders,partsupp,part,customer,sup
plier,nation,region');
    push (@defparams, 'view_views=mav_li,mv_locs');
    push (@defparams, 'viewlog_list=lineitem,orders');
    push (@defparams,
'mav_li_columns=count_p_group,l_shipdate,l_returnflag,
l_linestatus,sum_qty,sum_base_price ,sum_disc_price
,count_disc_price,sum_charge,
count_charge,count_qty,count_price,sum_disc,count_disc
,count_order');
    push (@defparams,
'tab_lineitem_columns=l_orderkey,l_partkey,l_suppkey,l
_linenumber,l_quantity,l_extendedprice,l_discount,l_ta
x,l_returnflag,l_linestatus,l_shipdate,l_commitdate,l_
receiptdate,l_shipinstruct,l_shipmode,l_comment');
    push (@defparams,
'tab_orders_columns=o_orderkey,o_custkey,o_orderstatus
,o_totalprice,o_orderdate,o_orderpriority,o_clerk,o_sh
ippriority,o_comment');
    push (@defparams,
'tab_partsupp_columns=ps_partkey,ps_suppkey,ps_availqt
y,ps_supplycost,ps_comment');
    push (@defparams,
'tab_part_columns=p_partkey,p_name,p_mfgr,p_brand,p_ty
pe,p_size,p_container,p_retailprice,p_comment');
    push (@defparams,
'tab_customer_columns=c_custkey,c_name,c_address,c_nat
ionkey,c_phone,c_acctbal,c_mktsegment,c_comment');
    push (@defparams,
'tab_supplier_columns=s_suppkey,s_name,s_address,s_nat
ionkey,s_phone,s_acctbal,s_comment');
    push (@defparams,
'tab_nation_columns=n_nationkey,n_name,n_regionkey,n_c
omment');
    push (@defparams,
'tab_region_columns=r_regionkey,r_name,r_comment');
    push (@defparams, 'tab_lineitem_#rows=6100000'); #
Somewhat of a random value
    push (@defparams, 'tab_orders_#rows=1500000');
    push (@defparams, 'tab_partsupp_#rows=800000');
    push (@defparams, 'tab_part_#rows=200000');
    push (@defparams, 'tab_customer_#rows=150000');
    push (@defparams, 'tab_supplier_#rows=10000');
    push (@defparams, 'tab_nation_#rows=0');
    push (@defparams, 'tab_region_#rows=0');
    push (@defparams, 'tab_lineitem_parttype=range');
    push (@defparams,
'tab_lineitem_l_orderkey_type=number');
    push (@defparams,
'tab_lineitem_l_partkey_type=number');
    push (@defparams,
'tab_lineitem_l_suppkey_type=number');
    push (@defparams,
'tab_lineitem_l_linenumber_type=number');
    push (@defparams,
'tab_lineitem_l_quantity_type=number');
    push (@defparams,
'tab_lineitem_l_extendedprice_type=number');
    push (@defparams,
'tab_lineitem_l_discount_type=number');
    push (@defparams,
'tab_lineitem_l_tax_type=number');
    push (@defparams,
'tab_lineitem_l_returnflag_type=char(1)');
    push (@defparams,
'tab_lineitem_l_linestatus_type=char(1)');
    push (@defparams,
'tab_lineitem_l_shipdate_type=date');
    push (@defparams,
'tab_lineitem_l_commitdate_type=date');
    push (@defparams,
'tab_lineitem_l_receiptdate_type=date');
    push (@defparams,
'tab_lineitem_l_shipinstruct_type=char(25)');
    push (@defparams,
'tab_lineitem_l_shipmode_type=char(10)');
    push (@defparams,
'tab_lineitem_l_comment_type=varchar(44)');
    push (@defparams,
'tab_lineitem_l_orderkey_pos=13:24');
    push (@defparams,
'tab_lineitem_l_partkey_pos=25:36');
    push (@defparams,
'tab_lineitem_l_suppkey_pos=37:48');
    push (@defparams,
'tab_lineitem_l_linenumber_pos=49:60');
    push (@defparams,
'tab_lineitem_l_quantity_pos=61:72');
    push (@defparams,
'tab_lineitem_l_extendedprice_pos=73:87');
    push (@defparams,
'tab_lineitem_l_discount_pos=88:102');
    push (@defparams,
'tab_lineitem_l_tax_pos=103:117');
    push (@defparams,
'tab_lineitem_l_returnflag_pos=118:118');
    push (@defparams,
'tab_lineitem_l_linestatus_pos=120:120');
    push (@defparams,
'tab_lineitem_l_shipdate_pos=122:131');
    push (@defparams,
'tab_lineitem_l_commitdate_pos=135:144');
    push (@defparams,
'tab_lineitem_l_receiptdate_pos=148:157');
    push (@defparams,
'tab_lineitem_l_shipinstruct_pos=161:185');
    push (@defparams,
'tab_lineitem_l_shipmode_pos=186:195');
    push (@defparams,
'tab_lineitem_l_comment_pos=196:239');
    push (@defparams,
'tab_lineitem_l_shipdate_loadcolx=date "yyyy-mm-dd"');
    push (@defparams,
'tab_lineitem_l_commitdate_loadcolx=date "yyyy-mm-
dd"');
    push (@defparams,
'tab_lineitem_l_receiptdate_loadcolx=date "yyyy-mm-
dd"');
    push (@defparams, 'tab_orders_parttype=range');
    push (@defparams,
'tab_orders_o_orderkey_type=number');
    push (@defparams,
'tab_orders_o_custkey_type=number');
    push (@defparams,
'tab_orders_o_orderstatus_type=char(1)');
    push (@defparams,
'tab_orders_o_totalprice_type=number');
```

```
    push (@defparams,
'tab_orders_o_orderdate_type=date');
    push (@defparams,
'tab_orders_o_orderpriority_type=char(15)');
    push (@defparams,
'tab_orders_o_clerk_type=char(15)');
    push (@defparams,
'tab_orders_o_shippriority_type=number');
    push (@defparams,
'tab_orders_o_comment_type=varchar(79)');
    push (@defparams,
'tab_orders_o_orderkey_pos=13:24');
    push (@defparams,
'tab_orders_o_custkey_pos=25:36');
    push (@defparams,
'tab_orders_o_orderstatus_pos=37:37');
    push (@defparams,
'tab_orders_o_totalprice_pos=39:53');
    push (@defparams,
'tab_orders_o_orderdate_pos=54:63');
    push (@defparams,
'tab_orders_o_orderpriority_pos=67:81');
    push (@defparams, 'tab_orders_o_clerk_pos=82:96');
    push (@defparams,
'tab_orders_o_shippriority_pos=97:108');
    push (@defparams,
'tab_orders_o_comment_pos=109:187');
    push (@defparams,
'tab_orders_o_orderdate_loadcolx=date "yyyy-mm-dd"');
    push (@defparams, 'tab_partsupp_parttype=range');
    push (@defparams,
'tab_partsupp_ps_partkey_type=number');
    push (@defparams,
'tab_partsupp_ps_suppkey_type=number');
    push (@defparams,
'tab_partsupp_ps_availqty_type=number');
    push (@defparams,
'tab_partsupp_ps_supplycost_type=number');
    push (@defparams,
'tab_partsupp_ps_comment_type=varchar(199)');
    push (@defparams,
'tab_partsupp_ps_partkey_pos=1:12');
    push (@defparams,
'tab_partsupp_ps_suppkey_pos=13:24');
    push (@defparams,
'tab_partsupp_ps_availqty_pos=25:36');
    push (@defparams,
'tab_partsupp_ps_supplycost_pos=37:51');
    push (@defparams,
'tab_partsupp_ps_comment_pos=52:250');
    push (@defparams, 'tab_part_parttype=range');
    push (@defparams,
'tab_part_p_partkey_type=number');
    push (@defparams,
'tab_part_p_name_type=varchar(55)');
    push (@defparams,
'tab_part_p_mfgr_type=char(25)');
    push (@defparams,
'tab_part_p_brand_type=char(10)');
    push (@defparams,
'tab_part_p_type_type=varchar(25)');
    push (@defparams, 'tab_part_p_size_type=number');
    push (@defparams,
'tab_part_p_container_type=char(10)');
    push (@defparams,
'tab_part_p_retailprice_type=number');
    push (@defparams,
'tab_part_p_comment_type=varchar(23)');
    push (@defparams, 'tab_part_p_partkey_pos=1:12');
    push (@defparams, 'tab_part_p_name_pos=13:67');
    push (@defparams, 'tab_part_p_mfgr_pos=68:92');
    push (@defparams, 'tab_part_p_brand_pos=93:102');
    push (@defparams, 'tab_part_p_type_pos=103:127');
    push (@defparams, 'tab_part_p_size_pos=128:139');
    push (@defparams,
'tab_part_p_container_pos=140:149');
    push (@defparams,
'tab_part_p_retailprice_pos=150:164');
    push (@defparams,
'tab_part_p_comment_pos=165:187');
    push (@defparams, 'tab_customer_parttype=range');
    push (@defparams,
'tab_customer_c_custkey_type=number');
    push (@defparams,
'tab_customer_c_name_type=varchar(25)');
    push (@defparams,
'tab_customer_c_address_type=varchar(40)');
    push (@defparams,
'tab_customer_c_nationkey_type=number');
    push (@defparams,
'tab_customer_c_phone_type=char(15)');
    push (@defparams,
'tab_customer_c_acctbal_type=number');
    push (@defparams,
'tab_customer_c_mktsegment_type=char(10)');
    push (@defparams,
'tab_customer_c_comment_type=varchar(117)');
    push (@defparams,
'tab_customer_c_custkey_pos=1:12');
    push (@defparams,
'tab_customer_c_name_pos=13:30');
    push (@defparams,
'tab_customer_c_address_pos=31:70');
    push (@defparams,
'tab_customer_c_nationkey_pos=71:82');
    push (@defparams,
'tab_customer_c_phone_pos=83:97');
    push (@defparams,
'tab_customer_c_acctbal_pos=98:112');
    push (@defparams,
'tab_customer_c_mktsegment_pos=113:122');
    push (@defparams,
'tab_customer_c_comment_pos=123:239');
    push (@defparams, 'tab_supplier_parttype=range');
    push (@defparams,
'tab_supplier_s_suppkey_type=number');
    push (@defparams,
'tab_supplier_s_name_type=char(25)');
    push (@defparams,
'tab_supplier_s_address_type=varchar(40)');
    push (@defparams,
'tab_supplier_s_nationkey_type=number');
    push (@defparams,
'tab_supplier_s_phone_type=char(15)');
    push (@defparams,
'tab_supplier_s_acctbal_type=number');
    push (@defparams,
'tab_supplier_s_comment_type=varchar(101)');
    push (@defparams,
'tab_supplier_s_suppkey_pos=1:12');
    push (@defparams,
'tab_supplier_s_name_pos=13:37');
    push (@defparams,
'tab_supplier_s_address_pos=38:77');
    push (@defparams,
'tab_supplier_s_nationkey_pos=78:89');
    push (@defparams,
'tab_supplier_s_phone_pos=90:104');
    push (@defparams,
'tab_supplier_s_acctbal_pos=105:119');
    push (@defparams,
'tab_supplier_s_comment_pos=120:220');
    push (@defparams, 'tab_nation_parttype=range');
    push (@defparams,
'tab_nation_n_nationkey_type=number');
    push (@defparams,
'tab_nation_n_name_type=char(25)');
    push (@defparams,
'tab_nation_n_regionkey_type=number');
    push (@defparams,
'tab_nation_n_comment_type=varchar(152)');
    push (@defparams,
'tab_nation_n_nationkey_pos=1:12');
    push (@defparams, 'tab_nation_n_name_pos=13:37');
    push (@defparams,
'tab_nation_n_regionkey_pos=38:49');
    push (@defparams,
'tab_nation_n_comment_pos=50:164');
    push (@defparams,
'tab_region_r_regionkey_type=number');
    push (@defparams,
'tab_region_r_name_type=char(25)');
    push (@defparams,
'tab_region_r_comment_type=varchar(152)');
    push (@defparams,
'tab_region_r_regionkey_pos=1:12');
    push (@defparams, 'tab_region_r_name_pos=13:37');
    push (@defparams,
'tab_region_r_comment_pos=38:152');
    push (@defparams,
'load_dbgen_lineitem_output_prefix='.'lineitem');
    push (@defparams,
'load_dbgen_orders_output_prefix='.'orders');
    push (@defparams,
'load_dbgen_customer_output_prefix='.'customer');
    push (@defparams,
'load_dbgen_parts_output_prefix='.'part');
    push (@defparams,
'load_dbgen_partsupp_output_prefix='.'partsupp');
    push (@defparams,
```

```
'load_dbgen_supplier_output_prefix='.'supplier');
    push (@defparams,
'load_dbgen_nation_output_prefix='.'nation');
    push (@defparams,
'load_dbgen_region_output_prefix='.'region');
    push (@defparams, 'load_type=fixed');
}


sub wisc
{
    @defparams = ();
    push (@defparams, 'user=wisc');
    push (@defparams, 'passwd=wisc');
    push (@defparams, 'tab_tables=wisc');
    push (@defparams,
'load_field_terminator=whitespace');
    push (@defparams,
'tab_wisc_columns=unique1,unique2,two,four,ten,twenty,
hundred,thousand,fivethous,tenthous,odd100,even100,str
ingu1,stringu2,string4');
    push (@defparams, 'tab_wisc_#rows=1000000');
    push (@defparams, 'tab_wisc_unique1_type=number');
    push (@defparams, 'tab_wisc_unique2_type=number');
    push (@defparams, 'tab_wisc_two_type=number');
    push (@defparams, 'tab_wisc_four_type=number');
    push (@defparams, 'tab_wisc_ten_type=number');
    push (@defparams, 'tab_wisc_twenty_type=number');
    push (@defparams, 'tab_wisc_hundred_type=number');
    push (@defparams,
'tab_wisc_thousand_type=number');
    push (@defparams,
'tab_wisc_fivethous_type=number');
    push (@defparams,
'tab_wisc_tenthous_type=number');
    push (@defparams, 'tab_wisc_odd100_type=number');
    push (@defparams, 'tab_wisc_even100_type=number');
    push (@defparams,
'tab_wisc_stringu1_type=varchar(52)');
    push (@defparams,
'tab_wisc_stringu2_type=varchar(52)');
    push (@defparams,
'tab_wisc_string4_type=varchar(52)');
    push (@defparams,
'tab_wisc_unique1_loadcolx=integer external');
    push (@defparams,
'tab_wisc_unique2_loadcolx=integer external');
    push (@defparams, 'tab_wisc_two_loadcolx=integer
external');
    push (@defparams, 'tab_wisc_four_loadcolx=integer
external');
    push (@defparams, 'tab_wisc_ten_loadcolx=integer
external');
    push (@defparams,
'tab_wisc_twenty_loadcolx=integer external');
    push (@defparams,
'tab_wisc_hundred_loadcolx=integer external');
    push (@defparams,
'tab_wisc_thousand_loadcolx=integer external');
    push (@defparams,
'tab_wisc_fivethous_loadcolx=integer external');
    push (@defparams,
'tab_wisc_tenthous_loadcolx=integer external');
    push (@defparams,
'tab_wisc_odd100_loadcolx=integer external');
    push (@defparams,
'tab_wisc_even100_loadcolx=integer external');
    push (@defparams,
'tab_wisc_stringu1_loadcolx=char(52)');
    push (@defparams,
'tab_wisc_stringu2_loadcolx=char(52)');
    push (@defparams,
'tab_wisc_string4_loadcolx=char(52)');
}

sub expand_filename
{
    # 1. parameter: name of datafiles in associative
array (params)
    # 2. parameter: number of files the name should be
expanded to
    @tsdf = split(/,/, $params{$_[0]});
    $no_files = $_[1];
    $cntr = 1;
    if ($tsdf[0] =~ /\#/) # we want to replace all #'s
    {
        $filenm = shift(@tsdf);
        $savename = $filenm;
        $params{$_[0]} = "";
        for ($i = 0; $i < $no_files; $i++)
        {
            $filenm =~ s/\#/$cntr/g;
            $cntr++;
            (($i+1) == $no_files) ? ($addcomma = "") :
($addcomma = ",");
            $params{$_[0]} .= $filenm . $addcomma;
            $filenm = $savename;
        }
        printf ("Expanded $savename
to:\n$params{$_[1]}\n\n") if $verbose;
    }
    else
    {
        if (@tsdf != $no_files)
        {
            print "Number of files (@ts_datafiles) for
$_[1] doesn't match\n _datafile parameter
$no_files}\n";
            exit(-1);
        }
    }
}

sub time0
{
    if (($os cmp "nt") == 0)
        {
            $value = shift (@_);
            $value = "*time=" . $value;
            $value = $value . "\n" if ($value !~
/.*\n$/);
            print OUTFILE "$value";
            $value = "";
        }
}

sub read_parameter_description
{
    @paramdesc = ();
    stat ("$pdfile");
    if (-e _)
    {
        open (PDFILE, "$pdfile");
        while (<PDFILE>)
        {
            ($key,$desc)=split (/::/,$_,2);
#           $key = shift(@l);
            $key =~ s/(\w*)<(\w*)>(\w*)/$1\\w\*$3/g;
            $paramdesc{$key} = $desc;
        }
    }
    else
    {
        print "Parameterfile: $pdfile does not exist\n
will continue without ...\n";
    }
}

sub key_exists
{
    $result = 1;
    return if ($runsilent == 1);
    @keys = keys %paramdesc;
    $p = $_[0]."!";
    while ($#keys >=0)
    {
        $key = pop(@keys);
        $key =$key."!";
        return if ($p =~ /$key/);
    }
    $result = -1;
}
```

==========

3tera_dbcre.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
```

```
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh

*load2
\sqlldr {}

*load1
\sqlldr {}
*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=64
############################################################
#######################
# Database Creation Phase
*sql
{
shutdown abort;
}
*wait
# creating database and initial rollback segment
*sql
{
startup pfile=
/export/home/oracle/oracle817/dbs/build_db1.ora
nomount;
create database
  controlfile reuse
  logfile
'/export/home/oracle/oracle817/dbs/datafiles/t3log1'
size 32000m reuse,

'/export/home/oracle/oracle817/dbs/datafiles/t3log2'
size 32000m reuse
  datafile
'/export/home/oracle/oracle817/dbs/datafiles/sys_1'
size 1000m reuse
  maxdatafiles 10000
  maxinstances 2
;

create public rollback segment t_rs1   storage
(initial 1m next 1m minextents 150 maxextents
unlimited);

alter rollback segment t_rs1 online;

shutdown
}
*wait
*sql
{
startup
pfile=/export/home/oracle/oracle817/dbs/build_db1.ora
}
*wait
*sql
{
create tablespace ts_undo_rep1
  datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_1_re
p1' size 16000m ;
}
*wait
# adding tpcd's ts_undo_rep1 datafiles
*sql
{
alter tablespace ts_undo_rep1
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_2_re
p1' size 16000m ;
}
*sql
{
alter tablespace ts_undo_rep1
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_3_re
p1' size 16000m ;
}
*sql
{
alter tablespace ts_undo_rep1
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_4_re
p1' size 16000m ;
}
*wait
# creating extra rollback segments
*sql
{
create rollback segment rrep11 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep12 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep13 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep14 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep15 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep16 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep17 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep18 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep19 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep110 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep111 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep112 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep113 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep114 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep115 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep116 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep117 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep118 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep119 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep120 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep121 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep122 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep123 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep124 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep125 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep126 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep127 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep128 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep129 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep130 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep131 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep132 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep133 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep134 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep135 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep136 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep137 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep138 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep139 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep140 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep141 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep142 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep143 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep144 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep145 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep146 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep147 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep148 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep149 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep150 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep151 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep152 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep153 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep154 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep155 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep156 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep157 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep158 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep159 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep160 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep161 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep162 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep163 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep164 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep165 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep166 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep167 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep168 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep169 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep170 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep171 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep172 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep173 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep174 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep175 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep176 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep177 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep178 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep179 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep180 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep181 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep182 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep183 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep184 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep185 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep186 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep187 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep188 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep189 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep190 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep191 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep192 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep193 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep194 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep195 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep196 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep197 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep198 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep199 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1100 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1101 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1102 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1103 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1104 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1105 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1106 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1107 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1108 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1109 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1110 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1111 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1112 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1113 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1114 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1115 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1116 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1117 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1118 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1119 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1120 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1121 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1122 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1123 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1124 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1125 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1126 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1127 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1128 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1129 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1130 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1131 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1132 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1133 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1134 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1135 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1136 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1137 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1138 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1139 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1140 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1141 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1142 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1143 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1144 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1145 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1146 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1147 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1148 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1149 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1150 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1151 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1152 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1153 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1154 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1155 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1156 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1157 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1158 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1159 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1160 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1161 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1162 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1163 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1164 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1165 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1166 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1167 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1168 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1169 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1170 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1171 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1172 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1173 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1174 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1175 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1176 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1177 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1178 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1179 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1180 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1181 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1182 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1183 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1184 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1185 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1186 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1187 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1188 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1189 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1190 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1191 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1192 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1193 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1194 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1195 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1196 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1197 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1198 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1199 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1200 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1201 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1202 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1203 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1204 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1205 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1206 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1207 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1208 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1209 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1210 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1211 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1212 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1213 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1214 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1215 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1216 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1217 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1218 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1219 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1220 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1221 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1222 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1223 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1224 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1225 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1226 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1227 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1228 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1229 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1230 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1231 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1232 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1233 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1234 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1235 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1236 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1237 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1238 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1239 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1240 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1241 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1242 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1243 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1244 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1245 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1246 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1247 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1248 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1249 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1250 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1251 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1252 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1253 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1254 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1255 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1256 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1257 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1258 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1259 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1260 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1261 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1262 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1263 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1264 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1265 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1266 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1267 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1268 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1269 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1270 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1271 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1272 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1273 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1274 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1275 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1276 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1277 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1278 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1279 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1280 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1281 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1282 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1283 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1284 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1285 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1286 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1287 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1288 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1289 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1290 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1291 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1292 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1293 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1294 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1295 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1296 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1297 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1298 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1299 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1300 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1301 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1302 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1303 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1304 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1305 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1306 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1307 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1308 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1309 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1310 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1311 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1312 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1313 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1314 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1315 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1316 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1317 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1318 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1319 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1320 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1321 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1322 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1323 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1324 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1325 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1326 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1327 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1328 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1329 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1330 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1331 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1332 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1333 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1334 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1335 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1336 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1337 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1338 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1339 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1340 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1341 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1342 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1343 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1344 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1345 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1346 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1347 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1348 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1349 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1350 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1351 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1352 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1353 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1354 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1355 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1356 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1357 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1358 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1359 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1360 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1361 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1362 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1363 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1364 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1365 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1366 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1367 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1368 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1369 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1370 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1371 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1372 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1373 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1374 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1375 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep1;
create rollback segment rrep1376 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1377 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1378 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1379 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1380 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1381 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1382 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1383 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1384 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1385 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1386 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1387 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1388 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1389 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1390 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1391 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1392 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1393 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1394 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1395 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1396 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1397 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1398 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1399 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
create rollback segment rrep1400 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep1;
}
*wait
*sql
{
create tablespace ts_undo_rep2
  datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_1_re
p2' size 16000m ;
}
*wait
# adding tpcd's ts_undo_rep2 datafiles
*sql
{
alter tablespace ts_undo_rep2
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_2_re
p2' size 16000m ;
}
*sql
{
alter tablespace ts_undo_rep2
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_3_re
p2' size 16000m ;
}
*sql
{
alter tablespace ts_undo_rep2
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/undo_4_re
p2' size 16000m ;
}
*wait
# creating extra rollback segments
*sql
{
create rollback segment rrep21 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep22 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep23 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep24 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep25 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep26 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep27 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep28 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep29 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep210 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep211 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep212 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep213 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep214 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep215 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep216 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep217 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep218 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep219 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep220 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep221 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep222 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep223 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep224 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep225 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep226 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep227 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep228 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep229 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep230 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep231 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep232 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep233 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep234 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep235 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep236 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep237 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep238 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep239 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep240 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep241 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep242 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep243 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep244 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep245 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep246 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep247 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep248 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep249 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep250 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep251 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep252 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep253 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep254 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep255 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep256 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep257 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep258 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep259 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep260 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep261 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep262 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep263 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep264 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep265 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep266 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep267 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep268 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep269 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep270 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep271 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep272 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep273 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep274 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep275 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep276 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep277 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep278 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep279 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep280 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep281 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep282 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep283 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep284 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep285 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep286 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep287 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep288 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep289 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep290 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep291 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep292 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep293 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep294 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep295 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep296 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep297 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep298 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep299 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2100 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2101 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2102 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2103 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2104 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2105 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2106 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2107 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2108 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2109 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2110 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2111 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2112 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2113 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2114 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2115 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2116 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2117 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2118 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2119 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2120 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2121 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2122 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2123 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2124 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2125 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2126 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2127 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2128 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2129 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2130 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2131 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2132 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2133 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2134 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2135 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2136 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2137 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2138 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2139 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2140 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2141 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2142 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2143 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2144 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2145 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2146 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2147 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2148 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2149 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2150 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2151 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2152 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2153 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2154 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2155 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2156 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2157 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2158 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2159 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2160 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2161 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2162 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2163 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2164 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2165 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2166 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2167 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2168 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2169 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2170 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2171 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2172 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2173 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2174 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2175 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2176 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2177 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2178 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2179 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2180 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2181 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2182 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2183 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2184 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2185 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2186 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2187 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2188 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2189 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2190 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2191 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2192 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2193 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2194 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2195 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2196 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2197 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2198 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2199 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2200 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2201 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2202 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2203 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2204 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2205 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2206 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2207 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2208 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2209 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2210 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2211 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2212 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2213 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2214 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2215 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2216 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2217 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2218 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2219 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2220 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2221 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2222 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2223 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2224 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2225 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2226 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2227 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2228 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2229 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2230 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2231 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2232 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2233 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2234 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2235 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2236 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2237 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2238 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2239 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2240 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2241 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2242 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2243 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2244 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2245 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2246 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2247 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2248 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2249 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2250 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2251 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2252 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2253 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2254 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2255 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2256 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2257 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2258 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2259 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2260 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2261 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2262 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2263 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2264 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2265 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2266 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2267 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2268 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2269 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2270 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2271 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2272 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2273 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2274 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2275 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2276 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2277 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2278 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2279 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2280 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2281 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2282 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2283 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2284 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2285 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2286 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2287 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2288 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2289 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2290 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2291 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2292 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2293 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2294 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2295 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2296 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2297 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2298 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2299 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2300 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2301 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2302 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2303 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2304 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2305 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2306 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2307 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2308 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2309 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2310 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2311 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2312 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2313 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2314 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2315 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2316 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2317 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2318 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2319 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2320 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2321 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2322 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2323 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2324 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2325 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2326 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2327 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2328 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2329 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2330 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2331 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2332 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2333 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2334 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2335 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2336 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2337 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2338 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2339 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2340 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2341 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2342 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2343 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2344 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2345 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2346 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2347 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2348 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2349 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2350 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2351 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2352 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2353 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2354 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2355 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2356 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2357 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2358 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2359 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2360 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2361 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2362 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2363 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2364 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2365 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2366 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2367 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2368 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2369 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2370 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2371 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2372 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2373 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2374 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2375 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2376 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2377 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2378 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2379 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2380 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2381 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2382 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2383 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
```

```
tablespace ts_undo_rep2;
create rollback segment rrep2384 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2385 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2386 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2387 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2388 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2389 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2390 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2391 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2392 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2393 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2394 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2395 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2396 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2397 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2398 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2399 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
create rollback segment rrep2400 storage (initial 1m
next 1m minextents 150 maxextents unlimited)
tablespace ts_undo_rep2;
}
*wait
# creating extra logfile threads
*sql
{
alter database add logfile thread 2
  '/export/home/oracle/oracle817/dbs/datafiles/t3log3'
size 32000m reuse,
  '/export/home/oracle/oracle817/dbs/datafiles/t3log4'
size 32000m reuse;
alter database enable public thread 2;
}
*wait
# building data dictionary
*sql
{
set termout off
set echo off
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@/export/home/oracle/oracle817/sqlplus/admin/pupbld.sq
l;
}
*wait
*sql
{
shutdown;
}
*wait
*sql
{
startup
pfile=/export/home/oracle/oracle817/dbs/init_inst1.ora
}
*wait
*bgoff
%e-dbcre

================================================
```

3tera_sctso.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load1
\sqlldr {}

*load2
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-sctso
*bgon=64
##########################################################
######################
# Schema Creation Phase - datafiles only (no tables or
users)
# creating data tablespaces, datafiles
# creating tpcd's ts_default tablespace
*sql
```

```
{
drop tablespace ts_default including contents;
create tablespace ts_default
datafile
'/export/home/oracle/oracle817/dbs/datafiles/default_1
' size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_s tablespace
*sql
{
drop tablespace ts_s including contents;
create tablespace ts_s
datafile
'/export/home/oracle/oracle817/dbs/datafiles/s_1' size
6144m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_c tablespace
*sql
{
drop tablespace ts_c including contents;
create tablespace ts_c
datafile
'/export/home/oracle/oracle817/dbs/datafiles/c_1' size
15350m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_ps tablespace
*sql
{
drop tablespace ts_ps including contents;
create tablespace ts_ps
datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_1'
size 24568m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_p tablespace
*sql
{
drop tablespace ts_p including contents;
create tablespace ts_p
datafile
'/export/home/oracle/oracle817/dbs/datafiles/p_1' size
20478m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_o1 tablespace
*sql
{
drop tablespace ts_o1 including contents;
create tablespace ts_o1
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_1' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o2 tablespace
*sql
{
drop tablespace ts_o2 including contents;
create tablespace ts_o2
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_2' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o3 tablespace
*sql
{
drop tablespace ts_o3 including contents;
create tablespace ts_o3
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_3' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o4 tablespace
*sql
{
drop tablespace ts_o4 including contents;
create tablespace ts_o4
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_4' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o5 tablespace
*sql
{
drop tablespace ts_o5 including contents;
create tablespace ts_o5
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_5' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o6 tablespace
*sql
{
drop tablespace ts_o6 including contents;
create tablespace ts_o6
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_6' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o7 tablespace
*sql
{
drop tablespace ts_o7 including contents;
create tablespace ts_o7
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_7' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o8 tablespace
*sql
{
drop tablespace ts_o8 including contents;
create tablespace ts_o8
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_8' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o9 tablespace
*sql
{
drop tablespace ts_o9 including contents;
create tablespace ts_o9
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_9' size
8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o10 tablespace
*sql
{
drop tablespace ts_o10 including contents;
create tablespace ts_o10
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_10'
size 8190m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_o11 tablespace
*sql
{
drop tablespace ts_o11 including contents;
create tablespace ts_o11
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_11'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o12 tablespace
*sql
{
drop tablespace ts_o12 including contents;
create tablespace ts_o12
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_12'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o13 tablespace
*sql
{
drop tablespace ts_o13 including contents;
create tablespace ts_o13
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_13'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o14 tablespace
*sql
{
drop tablespace ts_o14 including contents;
create tablespace ts_o14
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_14'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o15 tablespace
*sql
{
drop tablespace ts_o15 including contents;
create tablespace ts_o15
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_15'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o16 tablespace
*sql
{
drop tablespace ts_o16 including contents;
create tablespace ts_o16
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_16'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o17 tablespace
*sql
{
drop tablespace ts_o17 including contents;
create tablespace ts_o17
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_17'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o18 tablespace
*sql
{
drop tablespace ts_o18 including contents;
create tablespace ts_o18
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_18'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o19 tablespace
*sql
{
drop tablespace ts_o19 including contents;
create tablespace ts_o19
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_19'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o20 tablespace
*sql
{
drop tablespace ts_o20 including contents;
create tablespace ts_o20
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_20'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o21 tablespace
*sql
{
drop tablespace ts_o21 including contents;
create tablespace ts_o21
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_21'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o22 tablespace
*sql
{
drop tablespace ts_o22 including contents;
create tablespace ts_o22
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_22'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o23 tablespace
*sql
{
drop tablespace ts_o23 including contents;
create tablespace ts_o23
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_23'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o24 tablespace
*sql
{
drop tablespace ts_o24 including contents;
create tablespace ts_o24
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_24'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o25 tablespace
*sql
{
drop tablespace ts_o25 including contents;
create tablespace ts_o25
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_25'
size 8190m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_o26 tablespace
*sql
{
drop tablespace ts_o26 including contents;
create tablespace ts_o26
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_26'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o27 tablespace
*sql
{
drop tablespace ts_o27 including contents;
create tablespace ts_o27
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_27'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o28 tablespace
*sql
{
drop tablespace ts_o28 including contents;
create tablespace ts_o28
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_28'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o29 tablespace
*sql
{
drop tablespace ts_o29 including contents;
create tablespace ts_o29
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_29'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o30 tablespace
*sql
{
drop tablespace ts_o30 including contents;
create tablespace ts_o30
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_30'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o31 tablespace
*sql
{
drop tablespace ts_o31 including contents;
create tablespace ts_o31
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_31'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o32 tablespace
*sql
{
drop tablespace ts_o32 including contents;
create tablespace ts_o32
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_32'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o33 tablespace
*sql
{
drop tablespace ts_o33 including contents;
create tablespace ts_o33
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_33'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o34 tablespace
*sql
{
drop tablespace ts_o34 including contents;
create tablespace ts_o34
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_34'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o35 tablespace
*sql
{
drop tablespace ts_o35 including contents;
create tablespace ts_o35
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_35'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o36 tablespace
*sql
{
drop tablespace ts_o36 including contents;
create tablespace ts_o36
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_36'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o37 tablespace
*sql
{
drop tablespace ts_o37 including contents;
create tablespace ts_o37
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_37'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o38 tablespace
*sql
{
drop tablespace ts_o38 including contents;
create tablespace ts_o38
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_38'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o39 tablespace
*sql
{
drop tablespace ts_o39 including contents;
create tablespace ts_o39
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_39'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o40 tablespace
*sql
{
drop tablespace ts_o40 including contents;
create tablespace ts_o40
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_40'
size 8190m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_o41 tablespace
*sql
{
drop tablespace ts_o41 including contents;
create tablespace ts_o41
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_41'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o42 tablespace
*sql
{
drop tablespace ts_o42 including contents;
create tablespace ts_o42
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_42'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o43 tablespace
*sql
{
drop tablespace ts_o43 including contents;
create tablespace ts_o43
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_43'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o44 tablespace
*sql
{
drop tablespace ts_o44 including contents;
create tablespace ts_o44
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_44'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o45 tablespace
*sql
{
drop tablespace ts_o45 including contents;
create tablespace ts_o45
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_45'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o46 tablespace
*sql
{
drop tablespace ts_o46 including contents;
create tablespace ts_o46
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_46'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o47 tablespace
*sql
{
drop tablespace ts_o47 including contents;
create tablespace ts_o47
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_47'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o48 tablespace
*sql
{
drop tablespace ts_o48 including contents;
create tablespace ts_o48
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_48'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o49 tablespace
*sql
{
drop tablespace ts_o49 including contents;
create tablespace ts_o49
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_49'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o50 tablespace
*sql
{
drop tablespace ts_o50 including contents;
create tablespace ts_o50
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_50'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o51 tablespace
*sql
{
drop tablespace ts_o51 including contents;
create tablespace ts_o51
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_51'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o52 tablespace
*sql
{
drop tablespace ts_o52 including contents;
create tablespace ts_o52
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_52'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o53 tablespace
*sql
{
drop tablespace ts_o53 including contents;
create tablespace ts_o53
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_53'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o54 tablespace
*sql
{
drop tablespace ts_o54 including contents;
create tablespace ts_o54
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_54'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o55 tablespace
*sql
{
drop tablespace ts_o55 including contents;
create tablespace ts_o55
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_55'
size 8190m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_o56 tablespace
*sql
{
drop tablespace ts_o56 including contents;
create tablespace ts_o56
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_56'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o57 tablespace
*sql
{
drop tablespace ts_o57 including contents;
create tablespace ts_o57
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_57'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o58 tablespace
*sql
{
drop tablespace ts_o58 including contents;
create tablespace ts_o58
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_58'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o59 tablespace
*sql
{
drop tablespace ts_o59 including contents;
create tablespace ts_o59
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_59'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o60 tablespace
*sql
{
drop tablespace ts_o60 including contents;
create tablespace ts_o60
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_60'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o61 tablespace
*sql
{
drop tablespace ts_o61 including contents;
create tablespace ts_o61
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_61'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o62 tablespace
*sql
{
drop tablespace ts_o62 including contents;
create tablespace ts_o62
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_62'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o63 tablespace
*sql
{
drop tablespace ts_o63 including contents;
create tablespace ts_o63
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_63'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o64 tablespace
*sql
{
drop tablespace ts_o64 including contents;
create tablespace ts_o64
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_64'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o65 tablespace
*sql
{
drop tablespace ts_o65 including contents;
create tablespace ts_o65
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_65'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o66 tablespace
*sql
{
drop tablespace ts_o66 including contents;
create tablespace ts_o66
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_66'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o67 tablespace
*sql
{
drop tablespace ts_o67 including contents;
create tablespace ts_o67
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_67'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o68 tablespace
*sql
{
drop tablespace ts_o68 including contents;
create tablespace ts_o68
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_68'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o69 tablespace
*sql
{
drop tablespace ts_o69 including contents;
create tablespace ts_o69
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_69'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o70 tablespace
*sql
{
drop tablespace ts_o70 including contents;
create tablespace ts_o70
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_70'
size 8190m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_o71 tablespace
*sql
{
drop tablespace ts_o71 including contents;
create tablespace ts_o71
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_71'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o72 tablespace
*sql
{
drop tablespace ts_o72 including contents;
create tablespace ts_o72
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_72'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o73 tablespace
*sql
{
drop tablespace ts_o73 including contents;
create tablespace ts_o73
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_73'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o74 tablespace
*sql
{
drop tablespace ts_o74 including contents;
create tablespace ts_o74
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_74'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o75 tablespace
*sql
{
drop tablespace ts_o75 including contents;
create tablespace ts_o75
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_75'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o76 tablespace
*sql
{
drop tablespace ts_o76 including contents;
create tablespace ts_o76
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_76'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o77 tablespace
*sql
{
drop tablespace ts_o77 including contents;
create tablespace ts_o77
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_77'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o78 tablespace
*sql
{
drop tablespace ts_o78 including contents;
create tablespace ts_o78
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_78'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o79 tablespace
*sql
{
drop tablespace ts_o79 including contents;
create tablespace ts_o79
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_79'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o80 tablespace
*sql
{
drop tablespace ts_o80 including contents;
create tablespace ts_o80
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_80'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o81 tablespace
*sql
{
drop tablespace ts_o81 including contents;
create tablespace ts_o81
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_81'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o82 tablespace
*sql
{
drop tablespace ts_o82 including contents;
create tablespace ts_o82
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_82'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o83 tablespace
*sql
{
drop tablespace ts_o83 including contents;
create tablespace ts_o83
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_83'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_o84 tablespace
*sql
{
drop tablespace ts_o84 including contents;
create tablespace ts_o84
datafile
'/export/home/oracle/oracle817/dbs/datafiles/o_84'
size 8190m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l1 tablespace
*sql
{
drop tablespace ts_l1 including contents;
create tablespace ts_l1
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_1' size
32760m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_l2 tablespace
*sql
{
drop tablespace ts_l2 including contents;
create tablespace ts_l2
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_2' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l3 tablespace
*sql
{
drop tablespace ts_l3 including contents;
create tablespace ts_l3
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_3' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l4 tablespace
*sql
{
drop tablespace ts_l4 including contents;
create tablespace ts_l4
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_4' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l5 tablespace
*sql
{
drop tablespace ts_l5 including contents;
create tablespace ts_l5
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_5' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l6 tablespace
*sql
{
drop tablespace ts_l6 including contents;
create tablespace ts_l6
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_6' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l7 tablespace
*sql
{
drop tablespace ts_l7 including contents;
create tablespace ts_l7
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_7' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l8 tablespace
*sql
{
drop tablespace ts_l8 including contents;
create tablespace ts_l8
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_8' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l9 tablespace
*sql
{
drop tablespace ts_l9 including contents;
create tablespace ts_l9
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_9' size
32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l10 tablespace
*sql
{
drop tablespace ts_l10 including contents;
create tablespace ts_l10
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_10'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l11 tablespace
*sql
{
drop tablespace ts_l11 including contents;
create tablespace ts_l11
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_11'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l12 tablespace
*sql
{
drop tablespace ts_l12 including contents;
create tablespace ts_l12
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_12'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l13 tablespace
*sql
{
drop tablespace ts_l13 including contents;
create tablespace ts_l13
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_13'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l14 tablespace
*sql
{
drop tablespace ts_l14 including contents;
create tablespace ts_l14
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_14'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l15 tablespace
*sql
{
drop tablespace ts_l15 including contents;
create tablespace ts_l15
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_15'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l16 tablespace
*sql
{
drop tablespace ts_l16 including contents;
create tablespace ts_l16
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_16'
size 32760m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_l17 tablespace
*sql
{
drop tablespace ts_l17 including contents;
create tablespace ts_l17
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_17'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l18 tablespace
*sql
{
drop tablespace ts_l18 including contents;
create tablespace ts_l18
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_18'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l19 tablespace
*sql
{
drop tablespace ts_l19 including contents;
create tablespace ts_l19
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_19'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l20 tablespace
*sql
{
drop tablespace ts_l20 including contents;
create tablespace ts_l20
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_20'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l21 tablespace
*sql
{
drop tablespace ts_l21 including contents;
create tablespace ts_l21
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_21'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l22 tablespace
*sql
{
drop tablespace ts_l22 including contents;
create tablespace ts_l22
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_22'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l23 tablespace
*sql
{
drop tablespace ts_l23 including contents;
create tablespace ts_l23
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_23'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l24 tablespace
*sql
{
drop tablespace ts_l24 including contents;
create tablespace ts_l24
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_24'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l25 tablespace
*sql
{
drop tablespace ts_l25 including contents;
create tablespace ts_l25
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_25'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l26 tablespace
*sql
{
drop tablespace ts_l26 including contents;
create tablespace ts_l26
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_26'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l27 tablespace
*sql
{
drop tablespace ts_l27 including contents;
create tablespace ts_l27
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_27'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l28 tablespace
*sql
{
drop tablespace ts_l28 including contents;
create tablespace ts_l28
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_28'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l29 tablespace
*sql
{
drop tablespace ts_l29 including contents;
create tablespace ts_l29
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_29'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l30 tablespace
*sql
{
drop tablespace ts_l30 including contents;
create tablespace ts_l30
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_30'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l31 tablespace
*sql
{
drop tablespace ts_l31 including contents;
create tablespace ts_l31
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_31'
size 32760m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_l32 tablespace
*sql
{
drop tablespace ts_l32 including contents;
create tablespace ts_l32
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_32'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l33 tablespace
*sql
{
drop tablespace ts_l33 including contents;
create tablespace ts_l33
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_33'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l34 tablespace
*sql
{
drop tablespace ts_l34 including contents;
create tablespace ts_l34
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_34'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l35 tablespace
*sql
{
drop tablespace ts_l35 including contents;
create tablespace ts_l35
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_35'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l36 tablespace
*sql
{
drop tablespace ts_l36 including contents;
create tablespace ts_l36
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_36'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l37 tablespace
*sql
{
drop tablespace ts_l37 including contents;
create tablespace ts_l37
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_37'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l38 tablespace
*sql
{
drop tablespace ts_l38 including contents;
create tablespace ts_l38
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_38'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l39 tablespace
*sql
{
drop tablespace ts_l39 including contents;
create tablespace ts_l39
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_39'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l40 tablespace
*sql
{
drop tablespace ts_l40 including contents;
create tablespace ts_l40
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_40'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l41 tablespace
*sql
{
drop tablespace ts_l41 including contents;
create tablespace ts_l41
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_41'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l42 tablespace
*sql
{
drop tablespace ts_l42 including contents;
create tablespace ts_l42
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_42'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l43 tablespace
*sql
{
drop tablespace ts_l43 including contents;
create tablespace ts_l43
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_43'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l44 tablespace
*sql
{
drop tablespace ts_l44 including contents;
create tablespace ts_l44
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_44'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l45 tablespace
*sql
{
drop tablespace ts_l45 including contents;
create tablespace ts_l45
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_45'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l46 tablespace
*sql
{
drop tablespace ts_l46 including contents;
create tablespace ts_l46
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_46'
size 32760m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_l47 tablespace
*sql
{
drop tablespace ts_l47 including contents;
create tablespace ts_l47
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_47'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l48 tablespace
*sql
{
drop tablespace ts_l48 including contents;
create tablespace ts_l48
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_48'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l49 tablespace
*sql
{
drop tablespace ts_l49 including contents;
create tablespace ts_l49
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_49'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l50 tablespace
*sql
{
drop tablespace ts_l50 including contents;
create tablespace ts_l50
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_50'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l51 tablespace
*sql
{
drop tablespace ts_l51 including contents;
create tablespace ts_l51
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_51'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l52 tablespace
*sql
{
drop tablespace ts_l52 including contents;
create tablespace ts_l52
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_52'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l53 tablespace
*sql
{
drop tablespace ts_l53 including contents;
create tablespace ts_l53
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_53'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l54 tablespace
*sql
{
drop tablespace ts_l54 including contents;
create tablespace ts_l54
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_54'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l55 tablespace
*sql
{
drop tablespace ts_l55 including contents;
create tablespace ts_l55
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_55'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l56 tablespace
*sql
{
drop tablespace ts_l56 including contents;
create tablespace ts_l56
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_56'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l57 tablespace
*sql
{
drop tablespace ts_l57 including contents;
create tablespace ts_l57
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_57'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l58 tablespace
*sql
{
drop tablespace ts_l58 including contents;
create tablespace ts_l58
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_58'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l59 tablespace
*sql
{
drop tablespace ts_l59 including contents;
create tablespace ts_l59
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_59'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l60 tablespace
*sql
{
drop tablespace ts_l60 including contents;
create tablespace ts_l60
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_60'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l61 tablespace
*sql
{
drop tablespace ts_l61 including contents;
create tablespace ts_l61
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_61'
size 32760m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_l62 tablespace
*sql
{
drop tablespace ts_l62 including contents;
create tablespace ts_l62
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_62'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l63 tablespace
*sql
{
drop tablespace ts_l63 including contents;
create tablespace ts_l63
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_63'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l64 tablespace
*sql
{
drop tablespace ts_l64 including contents;
create tablespace ts_l64
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_64'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l65 tablespace
*sql
{
drop tablespace ts_l65 including contents;
create tablespace ts_l65
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_65'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l66 tablespace
*sql
{
drop tablespace ts_l66 including contents;
create tablespace ts_l66
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_66'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l67 tablespace
*sql
{
drop tablespace ts_l67 including contents;
create tablespace ts_l67
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_67'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l68 tablespace
*sql
{
drop tablespace ts_l68 including contents;
create tablespace ts_l68
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_68'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l69 tablespace
*sql
{
drop tablespace ts_l69 including contents;
create tablespace ts_l69
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_69'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l70 tablespace
*sql
{
drop tablespace ts_l70 including contents;
create tablespace ts_l70
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_70'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l71 tablespace
*sql
{
drop tablespace ts_l71 including contents;
create tablespace ts_l71
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_71'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l72 tablespace
*sql
{
drop tablespace ts_l72 including contents;
create tablespace ts_l72
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_72'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l73 tablespace
*sql
{
drop tablespace ts_l73 including contents;
create tablespace ts_l73
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_73'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l74 tablespace
*sql
{
drop tablespace ts_l74 including contents;
create tablespace ts_l74
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_74'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l75 tablespace
*sql
{
drop tablespace ts_l75 including contents;
create tablespace ts_l75
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_75'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l76 tablespace
*sql
{
drop tablespace ts_l76 including contents;
create tablespace ts_l76
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_76'
size 32760m reuse
extent management local
autoallocate
```

```
;
}
# creating tpcd's ts_l77 tablespace
*sql
{
drop tablespace ts_l77 including contents;
create tablespace ts_l77
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_77'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l78 tablespace
*sql
{
drop tablespace ts_l78 including contents;
create tablespace ts_l78
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_78'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l79 tablespace
*sql
{
drop tablespace ts_l79 including contents;
create tablespace ts_l79
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_79'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l80 tablespace
*sql
{
drop tablespace ts_l80 including contents;
create tablespace ts_l80
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_80'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l81 tablespace
*sql
{
drop tablespace ts_l81 including contents;
create tablespace ts_l81
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_81'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l82 tablespace
*sql
{
drop tablespace ts_l82 including contents;
create tablespace ts_l82
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_82'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l83 tablespace
*sql
{
drop tablespace ts_l83 including contents;
create tablespace ts_l83
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_83'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_l84 tablespace
*sql
{
drop tablespace ts_l84 including contents;
create tablespace ts_l84
datafile
'/export/home/oracle/oracle817/dbs/datafiles/l_84'
size 32760m reuse
extent management local
autoallocate
;
}
# creating tpcd's ts_i_lorderkey tablespace
*sql
{
drop tablespace ts_i_lorderkey including contents;
create tablespace ts_i_lorderkey
datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_1' size 32760m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_i_oorderkey tablespace
*sql
{
drop tablespace ts_i_oorderkey including contents;
create tablespace ts_i_oorderkey
datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_o_order
key_1' size 27640m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_i_ccustkey tablespace
*sql
{
drop tablespace ts_i_ccustkey including contents;
create tablespace ts_i_ccustkey
datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_c_custk
ey_1' size 27604m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_i_ps tablespace
*sql
{
drop tablespace ts_i_ps including contents;
create tablespace ts_i_ps
datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_ps_part
key_suppkey_1' size 27604m reuse
extent management local
autoallocate
 nologging
;
}
# creating tpcd's ts_temp tablespace
*sql
{
drop tablespace ts_temp including contents;
create temporary tablespace ts_temp
tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_1'
size 32760m reuse
extent management local
uniform size 100M
;
}
*wait
# adding tpcd's ts_default datafiles
# adding tpcd's ts_s datafiles
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/s_2' size
6144m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/s_3' size
6144m reuse;
}
*sql
```

```
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/s_4' size
6144m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/s_5' size
6144m reuse;
}
*sql
{
alter tablespace ts_s
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/s_6' size
6144m reuse;
}
# adding tpcd's ts_c datafiles
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/c_2' size
15350m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/c_3' size
15350m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/c_4' size
15350m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/c_5' size
15350m reuse;
}
*sql
{
alter tablespace ts_c
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/c_6' size
15350m reuse;
}
# adding tpcd's ts_ps datafiles
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_2'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_3'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_4'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_5'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_6'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_7'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_8'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_9'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_10'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_11'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_12'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_13'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_14'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_15'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_16'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_17'
size 24568m reuse;
}
*sql
{
alter tablespace ts_ps
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/ps_18'
size 24568m reuse;
}
# adding tpcd's ts_p datafiles
*sql
{
alter tablespace ts_p
```

```
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/p_2' size
20478m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/p_3' size
20478m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/p_4' size
20478m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/p_5' size
20478m reuse;
}
*sql
{
alter tablespace ts_p
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/p_6' size
20478m reuse;
}
# adding tpcd's ts_o1 datafiles
# adding tpcd's ts_o2 datafiles
# adding tpcd's ts_o3 datafiles
# adding tpcd's ts_o4 datafiles
# adding tpcd's ts_o5 datafiles
# adding tpcd's ts_o6 datafiles
# adding tpcd's ts_o7 datafiles
# adding tpcd's ts_o8 datafiles
# adding tpcd's ts_o9 datafiles
# adding tpcd's ts_o10 datafiles
# adding tpcd's ts_o11 datafiles
# adding tpcd's ts_o12 datafiles
# adding tpcd's ts_o13 datafiles
# adding tpcd's ts_o14 datafiles
# adding tpcd's ts_o15 datafiles
# adding tpcd's ts_o16 datafiles
# adding tpcd's ts_o17 datafiles
# adding tpcd's ts_o18 datafiles
# adding tpcd's ts_o19 datafiles
# adding tpcd's ts_o20 datafiles
# adding tpcd's ts_o21 datafiles
# adding tpcd's ts_o22 datafiles
# adding tpcd's ts_o23 datafiles
# adding tpcd's ts_o24 datafiles
# adding tpcd's ts_o25 datafiles
# adding tpcd's ts_o26 datafiles
# adding tpcd's ts_o27 datafiles
# adding tpcd's ts_o28 datafiles
# adding tpcd's ts_o29 datafiles
# adding tpcd's ts_o30 datafiles
# adding tpcd's ts_o31 datafiles
# adding tpcd's ts_o32 datafiles
# adding tpcd's ts_o33 datafiles
# adding tpcd's ts_o34 datafiles
# adding tpcd's ts_o35 datafiles
# adding tpcd's ts_o36 datafiles
# adding tpcd's ts_o37 datafiles
# adding tpcd's ts_o38 datafiles
# adding tpcd's ts_o39 datafiles
# adding tpcd's ts_o40 datafiles
# adding tpcd's ts_o41 datafiles
# adding tpcd's ts_o42 datafiles
# adding tpcd's ts_o43 datafiles
# adding tpcd's ts_o44 datafiles
# adding tpcd's ts_o45 datafiles
# adding tpcd's ts_o46 datafiles
# adding tpcd's ts_o47 datafiles
# adding tpcd's ts_o48 datafiles
# adding tpcd's ts_o49 datafiles
# adding tpcd's ts_o50 datafiles
# adding tpcd's ts_o51 datafiles
# adding tpcd's ts_o52 datafiles
# adding tpcd's ts_o53 datafiles
# adding tpcd's ts_o54 datafiles
# adding tpcd's ts_o55 datafiles
# adding tpcd's ts_o56 datafiles
# adding tpcd's ts_o57 datafiles
# adding tpcd's ts_o58 datafiles
# adding tpcd's ts_o59 datafiles
# adding tpcd's ts_o60 datafiles
# adding tpcd's ts_o61 datafiles
# adding tpcd's ts_o62 datafiles
# adding tpcd's ts_o63 datafiles
# adding tpcd's ts_o64 datafiles
# adding tpcd's ts_o65 datafiles
# adding tpcd's ts_o66 datafiles
# adding tpcd's ts_o67 datafiles
# adding tpcd's ts_o68 datafiles
# adding tpcd's ts_o69 datafiles
# adding tpcd's ts_o70 datafiles
# adding tpcd's ts_o71 datafiles
# adding tpcd's ts_o72 datafiles
# adding tpcd's ts_o73 datafiles
# adding tpcd's ts_o74 datafiles
# adding tpcd's ts_o75 datafiles
# adding tpcd's ts_o76 datafiles
# adding tpcd's ts_o77 datafiles
# adding tpcd's ts_o78 datafiles
# adding tpcd's ts_o79 datafiles
# adding tpcd's ts_o80 datafiles
# adding tpcd's ts_o81 datafiles
# adding tpcd's ts_o82 datafiles
# adding tpcd's ts_o83 datafiles
# adding tpcd's ts_o84 datafiles

# adding tpcd's ts_l1 datafiles
*sql
{
alter tablespace ts_l1
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_1'
size 5114m reuse;
}
# adding tpcd's ts_l2 datafiles
*sql
{
alter tablespace ts_l2
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_2'
size 5114m reuse;
}
# adding tpcd's ts_l3 datafiles
*sql
{
alter tablespace ts_l3
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_3'
size 5114m reuse;
}
# adding tpcd's ts_l4 datafiles
*sql
{
alter tablespace ts_l4
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_4'
size 5114m reuse;
}
# adding tpcd's ts_l5 datafiles
*sql
{
alter tablespace ts_l5
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_5'
size 5114m reuse;
}
# adding tpcd's ts_l6 datafiles
*sql
{
alter tablespace ts_l6
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_1'
size 5114m reuse;
}
# adding tpcd's ts_l7 datafiles
*sql
{
alter tablespace ts_l7
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_2'
size 5114m reuse;
}
# adding tpcd's ts_l8 datafiles
*sql
{
alter tablespace ts_l8
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_3'
size 5114m reuse;
```

```
}
# adding tpcd's ts_l9 datafiles
*sql
{
alter tablespace ts_l9
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_4'
size 5114m reuse;
}
# adding tpcd's ts_l10 datafiles
*sql
{
alter tablespace ts_l10
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_5'
size 5114m reuse;
}
# adding tpcd's ts_l11 datafiles
*sql
{
alter tablespace ts_l11
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_6'
size 5114m reuse;
}
# adding tpcd's ts_l12 datafiles
*sql
{
alter tablespace ts_l12
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_7'
size 5114m reuse;
}
# adding tpcd's ts_l13 datafiles
*sql
{
alter tablespace ts_l13
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_8'
size 5114m reuse;
}
# adding tpcd's ts_l14 datafiles
*sql
{
alter tablespace ts_l14
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_9'
size 5114m reuse;
}
# adding tpcd's ts_l15 datafiles
*sql
{
alter tablespace ts_l15
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_10'
size 5114m reuse;
}
# adding tpcd's ts_l16 datafiles
*sql
{
alter tablespace ts_l16
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_11'
size 5114m reuse;
}
# adding tpcd's ts_l17 datafiles
*sql
{
alter tablespace ts_l17
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_12'
size 5114m reuse;
}
# adding tpcd's ts_l18 datafiles
*sql
{
alter tablespace ts_l18
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_13'
size 5114m reuse;
}
# adding tpcd's ts_l19 datafiles
*sql
{
alter tablespace ts_l19
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_14'
size 5114m reuse;
}
# adding tpcd's ts_l20 datafiles
*sql
{
alter tablespace ts_l20
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_15'
size 5114m reuse;
}
# adding tpcd's ts_l21 datafiles
*sql
{
alter tablespace ts_l21
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_16'
size 5114m reuse;
}
# adding tpcd's ts_l22 datafiles
*sql
{
alter tablespace ts_l22
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_17'
size 5114m reuse;
}
# adding tpcd's ts_l23 datafiles
*sql
{
alter tablespace ts_l23
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_18'
size 5114m reuse;
}
# adding tpcd's ts_l24 datafiles
*sql
{
alter tablespace ts_l24
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_19'
size 5114m reuse;
}
# adding tpcd's ts_l25 datafiles
*sql
{
alter tablespace ts_l25
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_20'
size 5114m reuse;
}
# adding tpcd's ts_l26 datafiles
*sql
{
alter tablespace ts_l26
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_21'
size 5114m reuse;
}
# adding tpcd's ts_l27 datafiles
*sql
{
alter tablespace ts_l27
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_22'
size 5114m reuse;
}
# adding tpcd's ts_l28 datafiles
*sql
{
alter tablespace ts_l28
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_23'
size 5114m reuse;
}
# adding tpcd's ts_l29 datafiles
*sql
{
alter tablespace ts_l29
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_24'
size 5114m reuse;
}
# adding tpcd's ts_l30 datafiles
*sql
{
alter tablespace ts_l30
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_25'
size 5114m reuse;
}
# adding tpcd's ts_l31 datafiles
*sql
{
```

```
alter tablespace ts_l31
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_26'
size 5114m reuse;
}
# adding tpcd's ts_l32 datafiles
*sql
{
alter tablespace ts_l32
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_27'
size 5114m reuse;
}
# adding tpcd's ts_l33 datafiles
*sql
{
alter tablespace ts_l33
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_28'
size 5114m reuse;
}
# adding tpcd's ts_l34 datafiles
*sql
{
alter tablespace ts_l34
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_29'
size 5114m reuse;
}
# adding tpcd's ts_l35 datafiles
*sql
{
alter tablespace ts_l35
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_30'
size 5114m reuse;
}
# adding tpcd's ts_l36 datafiles
*sql
{
alter tablespace ts_l36
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_31'
size 5114m reuse;
}
# adding tpcd's ts_l37 datafiles
*sql
{
alter tablespace ts_l37
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_32'
size 5114m reuse;
}
# adding tpcd's ts_l38 datafiles
*sql
{
alter tablespace ts_l38
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_33'
size 5114m reuse;
}
# adding tpcd's ts_l39 datafiles
*sql
{
alter tablespace ts_l39
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_34'
size 5114m reuse;
}
# adding tpcd's ts_l40 datafiles
*sql
{
alter tablespace ts_l40
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_35'
size 5114m reuse;
}
# adding tpcd's ts_l41 datafiles
*sql
{
alter tablespace ts_l41
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_36'
size 5114m reuse;
}
# adding tpcd's ts_l42 datafiles
*sql
{
alter tablespace ts_l42
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_37'
size 5114m reuse;
}
# adding tpcd's ts_l43 datafiles
*sql
{
alter tablespace ts_l43
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_38'
size 5114m reuse;
}
# adding tpcd's ts_l44 datafiles
*sql
{
alter tablespace ts_l44
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_39'
size 5114m reuse;
}
# adding tpcd's ts_l45 datafiles
*sql
{
alter tablespace ts_l45
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_40'
size 5114m reuse;
}
# adding tpcd's ts_l46 datafiles
*sql
{
alter tablespace ts_l46
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_41'
size 5114m reuse;
}
# adding tpcd's ts_l47 datafiles
*sql
{
alter tablespace ts_l47
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_42'
size 5114m reuse;
}
# adding tpcd's ts_l48 datafiles
*sql
{
alter tablespace ts_l48
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_43'
size 5114m reuse;
}
# adding tpcd's ts_l49 datafiles
*sql
{
alter tablespace ts_l49
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_44'
size 5114m reuse;
}
# adding tpcd's ts_l50 datafiles
*sql
{
alter tablespace ts_l50
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_45'
size 5114m reuse;
}
# adding tpcd's ts_l51 datafiles
*sql
{
alter tablespace ts_l51
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_46'
size 5114m reuse;
}
# adding tpcd's ts_l52 datafiles
*sql
{
alter tablespace ts_l52
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_47'
size 5114m reuse;
}
# adding tpcd's ts_l53 datafiles
*sql
{
alter tablespace ts_l53
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_48'
size 5114m reuse;
```

```
}
# adding tpcd's ts_l54 datafiles
*sql
{
alter tablespace ts_l54
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_49'
size 5114m reuse;
}
# adding tpcd's ts_l55 datafiles
*sql
{
alter tablespace ts_l55
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_50'
size 5114m reuse;
}
# adding tpcd's ts_l56 datafiles
*sql
{
alter tablespace ts_l56
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_51'
size 5114m reuse;
}
# adding tpcd's ts_l57 datafiles
*sql
{
alter tablespace ts_l57
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_52'
size 5114m reuse;
}
# adding tpcd's ts_l58 datafiles
*sql
{
alter tablespace ts_l58
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_53'
size 5114m reuse;
}
# adding tpcd's ts_l59 datafiles
*sql
{
alter tablespace ts_l59
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_54'
size 5114m reuse;
}
# adding tpcd's ts_l60 datafiles
*sql
{
alter tablespace ts_l60
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_55'
size 5114m reuse;
}
# adding tpcd's ts_l61 datafiles
*sql
{
alter tablespace ts_l61
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_56'
size 5114m reuse;
}
# adding tpcd's ts_l62 datafiles
*sql
{
alter tablespace ts_l62
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_57'
size 5114m reuse;
}
# adding tpcd's ts_l63 datafiles
*sql
{
alter tablespace ts_l63
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_58'
size 5114m reuse;
}
# adding tpcd's ts_l64 datafiles
*sql
{
alter tablespace ts_l64
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_59'
size 5114m reuse;
}
# adding tpcd's ts_l65 datafiles
*sql
{
alter tablespace ts_l65
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_60'
size 5114m reuse;
}
# adding tpcd's ts_l66 datafiles
*sql
{
alter tablespace ts_l66
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_61'
size 5114m reuse;
}
# adding tpcd's ts_l67 datafiles
*sql
{
alter tablespace ts_l67
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_62'
size 5114m reuse;
}
# adding tpcd's ts_l68 datafiles
*sql
{
alter tablespace ts_l68
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_63'
size 5114m reuse;
}
# adding tpcd's ts_l69 datafiles
*sql
{
alter tablespace ts_l69
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_64'
size 5114m reuse;
}
# adding tpcd's ts_l70 datafiles
*sql
{
alter tablespace ts_l70
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_65'
size 5114m reuse;
}
# adding tpcd's ts_l71 datafiles
*sql
{
alter tablespace ts_l71
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_66'
size 5114m reuse;
}
# adding tpcd's ts_l72 datafiles
*sql
{
alter tablespace ts_l72
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_67'
size 5114m reuse;
}
# adding tpcd's ts_l73 datafiles
*sql
{
alter tablespace ts_l73
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_68'
size 5114m reuse;
}
# adding tpcd's ts_l74 datafiles
*sql
{
alter tablespace ts_l74
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_69'
size 5114m reuse;
}
# adding tpcd's ts_l75 datafiles
*sql
{
alter tablespace ts_l75
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_70'
size 5114m reuse;
}
# adding tpcd's ts_l76 datafiles
*sql
{
```

```
alter tablespace ts_l76
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_71'
size 5114m reuse;
}
# adding tpcd's ts_l77 datafiles
*sql
{
alter tablespace ts_l77
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_72'
size 5114m reuse;
}
# adding tpcd's ts_l78 datafiles
*sql
{
alter tablespace ts_l78
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_73'
size 5114m reuse;
}
# adding tpcd's ts_l79 datafiles
*sql
{
alter tablespace ts_l79
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_74'
size 5114m reuse;
}
# adding tpcd's ts_l80 datafiles
*sql
{
alter tablespace ts_l80
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_75'
size 5114m reuse;
}
# adding tpcd's ts_l81 datafiles
*sql
{
alter tablespace ts_l81
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li3_76'
size 5114m reuse;
}
# adding tpcd's ts_l82 datafiles
*sql
{
alter tablespace ts_l82
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_6'
size 5114m reuse;
}
# adding tpcd's ts_l83 datafiles
*sql
{
alter tablespace ts_l83
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_7'
size 5114m reuse;
}
# adding tpcd's ts_l84 datafiles
*sql
{
alter tablespace ts_l84
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/li4_8'
size 5114m reuse;
}
# adding tpcd's ts_i_lorderkey datafiles
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_2' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_3' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_4' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_5' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_6' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_7' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_8' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_9' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_10' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_11' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_12' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_13' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_14' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_15' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_16' size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
'/export/home/oracle/oracle817/dbs/datafiles/i_l_order
```

```
key_17’ size 32760m reuse;
}
*sql
{
alter tablespace ts_i_lorderkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_l_order
key_18’ size 32760m reuse;
}
# adding tpcd’s ts_i_oorderkey datafiles
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_o_order
key_2’ size 27640m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_o_order
key_3’ size 27640m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_o_order
key_4’ size 27640m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_o_order
key_5’ size 27640m reuse;
}
*sql
{
alter tablespace ts_i_oorderkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_o_order
key_6’ size 27640m reuse;
}
# adding tpcd’s ts_i_ccustkey datafiles
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_c_custk
ey_2’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_c_custk
ey_3’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_c_custk
ey_4’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_c_custk
ey_5’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ccustkey
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_c_custk
ey_6’ size 27604m reuse;
}
# adding tpcd’s ts_i_ps datafiles
*sql
{
alter tablespace ts_i_ps
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_ps_part
key_suppkey_2’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ps
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_ps_part
key_suppkey_3’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ps
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_ps_part
key_suppkey_4’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ps
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_ps_part
key_suppkey_5’ size 27604m reuse;
}
*sql
{
alter tablespace ts_i_ps
  add datafile
’/export/home/oracle/oracle817/dbs/datafiles/i_ps_part
key_suppkey_6’ size 27604m reuse;
}
# adding tpcd’s ts_temp datafiles
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_2’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_3’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_4’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_5’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_6’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_7’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_8’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_9’
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
’/export/home/oracle/oracle817/dbs/datafiles/temp_10’
size 32760m reuse;
```

```
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_11'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_12'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_13'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_14'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_15'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_16'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_17'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_18'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_19'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_20'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_21'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_22'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_23'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_24'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_25'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_26'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_27'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_28'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_29'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_30'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_31'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_32'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_33'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_34'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_35'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
```

```
'/export/home/oracle/oracle817/dbs/datafiles/temp_36'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_37'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_38'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_39'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_40'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_41'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_42'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_43'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_44'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_45'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_46'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_47'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_48'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_49'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_50'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_51'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_52'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_53'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_54'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_55'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_56'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_57'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_58'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_59'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_60'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_61'
size 32760m reuse;
}
*sql
{
```

```
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_62'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_63'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_64'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_65'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_66'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_67'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_68'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_69'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_70'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_71'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_72'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_73'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_74'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_75'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_76'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_77'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_78'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_79'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_80'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_81'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_82'
size 32760m reuse;
}
*sql
{
alter tablespace ts_temp
  add tempfile
'/export/home/oracle/oracle817/dbs/datafiles/temp_83'
size 32760m reuse;
}
*wait
*wait
*bgoff
%e-sctso
====================================================
```

3tera_scuto.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!
```

```
*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load1
\sqlldr {}

*load2
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-scuto
*bgon=64
############################################################
######################
# Schema Creation Phase - User and Tables ONLY (no
datafiles)
# creating tpcd user
*sql
{
drop user tpcd cascade;
grant DBA
  to tpcd identified by tpcd;
}
*wait
*sql
{
connect tpcd/tpcd;
}
*wait
*sql
{
connect tpcd/tpcd;
drop table lineitem;
create table lineitem(
    l_shipdate          date ,
    l_orderkey          number NOT NULL,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number NOT NULL,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number NOT NULL,
    l_linestatus        char(1) ,
    l_tax               number NOT NULL,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          char(10) ,
    l_linenumber        number NOT NULL,
    l_shipinstruct      char(25) ,
    l_comment           varchar(44)
)
pctfree 1
pctused 99
initrans 10
storage (freelists 99 freelist groups 2)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 16
(
partition item1 values less than (to_date('1992-01-
01','YYYY-MM-DD'))
tablespace ts_l1
,
partition item2 values less than (to_date('1992-02-
01','YYYY-MM-DD'))
tablespace ts_l2
,
partition item3 values less than (to_date('1992-03-
01','YYYY-MM-DD'))
tablespace ts_l3
,
partition item4 values less than (to_date('1992-04-
01','YYYY-MM-DD'))
tablespace ts_l4
,
partition item5 values less than (to_date('1992-05-
01','YYYY-MM-DD'))
tablespace ts_l5
,
partition item6 values less than (to_date('1992-06-
01','YYYY-MM-DD'))
tablespace ts_l6
,
partition item7 values less than (to_date('1992-07-
01','YYYY-MM-DD'))
tablespace ts_l7
,
partition item8 values less than (to_date('1992-08-
01','YYYY-MM-DD'))
tablespace ts_l8
,
partition item9 values less than (to_date('1992-09-
01','YYYY-MM-DD'))
tablespace ts_l9
,
partition item10 values less than (to_date('1992-10-
01','YYYY-MM-DD'))
tablespace ts_l10
,
partition item11 values less than (to_date('1992-11-
01','YYYY-MM-DD'))
tablespace ts_l11
,
partition item12 values less than (to_date('1992-12-
01','YYYY-MM-DD'))
tablespace ts_l12
,
partition item13 values less than (to_date('1993-01-
01','YYYY-MM-DD'))
tablespace ts_l13
,
partition item14 values less than (to_date('1993-02-
01','YYYY-MM-DD'))
tablespace ts_l14
,
partition item15 values less than (to_date('1993-03-
01','YYYY-MM-DD'))
tablespace ts_l15
,
partition item16 values less than (to_date('1993-04-
01','YYYY-MM-DD'))
tablespace ts_l16
,
partition item17 values less than (to_date('1993-05-
01','YYYY-MM-DD'))
tablespace ts_l17
,
partition item18 values less than (to_date('1993-06-
01','YYYY-MM-DD'))
tablespace ts_l18
,
partition item19 values less than (to_date('1993-07-
01','YYYY-MM-DD'))
tablespace ts_l19
```

```
,
partition item20 values less than (to_date('1993-08-
01','YYYY-MM-DD'))
tablespace ts_l20
,
partition item21 values less than (to_date('1993-09-
01','YYYY-MM-DD'))
tablespace ts_l21
,
partition item22 values less than (to_date('1993-10-
01','YYYY-MM-DD'))
tablespace ts_l22
,
partition item23 values less than (to_date('1993-11-
01','YYYY-MM-DD'))
tablespace ts_l23
,
partition item24 values less than (to_date('1993-12-
01','YYYY-MM-DD'))
tablespace ts_l24
,
partition item25 values less than (to_date('1994-01-
01','YYYY-MM-DD'))
tablespace ts_l25
,
partition item26 values less than (to_date('1994-02-
01','YYYY-MM-DD'))
tablespace ts_l26
,
partition item27 values less than (to_date('1994-03-
01','YYYY-MM-DD'))
tablespace ts_l27
,
partition item28 values less than (to_date('1994-04-
01','YYYY-MM-DD'))
tablespace ts_l28
,
partition item29 values less than (to_date('1994-05-
01','YYYY-MM-DD'))
tablespace ts_l29
,
partition item30 values less than (to_date('1994-06-
01','YYYY-MM-DD'))
tablespace ts_l30
,
partition item31 values less than (to_date('1994-07-
01','YYYY-MM-DD'))
tablespace ts_l31
,
partition item32 values less than (to_date('1994-08-
01','YYYY-MM-DD'))
tablespace ts_l32
,
partition item33 values less than (to_date('1994-09-
01','YYYY-MM-DD'))
tablespace ts_l33
,
partition item34 values less than (to_date('1994-10-
01','YYYY-MM-DD'))
tablespace ts_l34
,
partition item35 values less than (to_date('1994-11-
01','YYYY-MM-DD'))
tablespace ts_l35
,
partition item36 values less than (to_date('1994-12-
01','YYYY-MM-DD'))
tablespace ts_l36
,
partition item37 values less than (to_date('1995-01-
01','YYYY-MM-DD'))
tablespace ts_l37
,
partition item38 values less than (to_date('1995-02-
01','YYYY-MM-DD'))
tablespace ts_l38
,
partition item39 values less than (to_date('1995-03-
01','YYYY-MM-DD'))
tablespace ts_l39
,
partition item40 values less than (to_date('1995-04-
01','YYYY-MM-DD'))
tablespace ts_l40
,
partition item41 values less than (to_date('1995-05-
01','YYYY-MM-DD'))
tablespace ts_l41
,
partition item42 values less than (to_date('1995-06-
01','YYYY-MM-DD'))
tablespace ts_l42
,
partition item43 values less than (to_date('1995-07-
01','YYYY-MM-DD'))
tablespace ts_l43
,
partition item44 values less than (to_date('1995-08-
01','YYYY-MM-DD'))
tablespace ts_l44
,
partition item45 values less than (to_date('1995-09-
01','YYYY-MM-DD'))
tablespace ts_l45
,
partition item46 values less than (to_date('1995-10-
01','YYYY-MM-DD'))
tablespace ts_l46
,
partition item47 values less than (to_date('1995-11-
01','YYYY-MM-DD'))
tablespace ts_l47
,
partition item48 values less than (to_date('1995-12-
01','YYYY-MM-DD'))
tablespace ts_l48
,
partition item49 values less than (to_date('1996-01-
01','YYYY-MM-DD'))
tablespace ts_l49
,
partition item50 values less than (to_date('1996-02-
01','YYYY-MM-DD'))
tablespace ts_l50
,
partition item51 values less than (to_date('1996-03-
01','YYYY-MM-DD'))
tablespace ts_l51
,
partition item52 values less than (to_date('1996-04-
01','YYYY-MM-DD'))
tablespace ts_l52
,
partition item53 values less than (to_date('1996-05-
01','YYYY-MM-DD'))
tablespace ts_l53
,
partition item54 values less than (to_date('1996-06-
01','YYYY-MM-DD'))
tablespace ts_l54
,
partition item55 values less than (to_date('1996-07-
01','YYYY-MM-DD'))
tablespace ts_l55
,
partition item56 values less than (to_date('1996-08-
01','YYYY-MM-DD'))
tablespace ts_l56
,
partition item57 values less than (to_date('1996-09-
01','YYYY-MM-DD'))
tablespace ts_l57
,
partition item58 values less than (to_date('1996-10-
01','YYYY-MM-DD'))
tablespace ts_l58
,
partition item59 values less than (to_date('1996-11-
01','YYYY-MM-DD'))
tablespace ts_l59
,
partition item60 values less than (to_date('1996-12-
01','YYYY-MM-DD'))
tablespace ts_l60
,
partition item61 values less than (to_date('1997-01-
01','YYYY-MM-DD'))
tablespace ts_l61
,
partition item62 values less than (to_date('1997-02-
01','YYYY-MM-DD'))
tablespace ts_l62
,
partition item63 values less than (to_date('1997-03-
01','YYYY-MM-DD'))
tablespace ts_l63
,
partition item64 values less than (to_date('1997-04-
01','YYYY-MM-DD'))
tablespace ts_l64
```

```
,
partition item65 values less than (to_date('1997-05-
01','YYYY-MM-DD'))
tablespace ts_l65
,
partition item66 values less than (to_date('1997-06-
01','YYYY-MM-DD'))
tablespace ts_l66
,
partition item67 values less than (to_date('1997-07-
01','YYYY-MM-DD'))
tablespace ts_l67
,
partition item68 values less than (to_date('1997-08-
01','YYYY-MM-DD'))
tablespace ts_l68
,
partition item69 values less than (to_date('1997-09-
01','YYYY-MM-DD'))
tablespace ts_l69
,
partition item70 values less than (to_date('1997-10-
01','YYYY-MM-DD'))
tablespace ts_l70
,
partition item71 values less than (to_date('1997-11-
01','YYYY-MM-DD'))
tablespace ts_l71
,
partition item72 values less than (to_date('1997-12-
01','YYYY-MM-DD'))
tablespace ts_l72
,
partition item73 values less than (to_date('1998-01-
01','YYYY-MM-DD'))
tablespace ts_l73
,
partition item74 values less than (to_date('1998-02-
01','YYYY-MM-DD'))
tablespace ts_l74
,
partition item75 values less than (to_date('1998-03-
01','YYYY-MM-DD'))
tablespace ts_l75
,
partition item76 values less than (to_date('1998-04-
01','YYYY-MM-DD'))
tablespace ts_l76
,
partition item77 values less than (to_date('1998-05-
01','YYYY-MM-DD'))
tablespace ts_l77
,
partition item78 values less than (to_date('1998-06-
01','YYYY-MM-DD'))
tablespace ts_l78
,
partition item79 values less than (to_date('1998-07-
01','YYYY-MM-DD'))
tablespace ts_l79
,
partition item80 values less than (to_date('1998-08-
01','YYYY-MM-DD'))
tablespace ts_l80
,
partition item81 values less than (to_date('1998-09-
01','YYYY-MM-DD'))
tablespace ts_l81
,
partition item82 values less than (to_date('1998-10-
01','YYYY-MM-DD'))
tablespace ts_l82
,
partition item83 values less than (to_date('1998-11-
01','YYYY-MM-DD'))
tablespace ts_l83
,
partition item84 values less than (MAXVALUE)
tablespace ts_l84
)
;
drop table orders;
create table orders(
    o_orderdate          date ,
    o_orderkey           number NOT NULL,
    o_custkey            number NOT NULL,
    o_orderpriority      char(15) ,
    o_shippriority       number ,
    o_clerk              char(15) ,
    o_orderstatus        char(1) ,
    o_totalprice         number ,
    o_comment            varchar(79)
)
pctfree 1
pctused 99
initrans 10
storage (freelists 99 freelist groups 2)
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 16
(
partition ord1 values less than (to_date('1992-01-
01','YYYY-MM-DD'))
tablespace ts_o1
,
partition ord2 values less than (to_date('1992-02-
01','YYYY-MM-DD'))
tablespace ts_o2
,
partition ord3 values less than (to_date('1992-03-
01','YYYY-MM-DD'))
tablespace ts_o3
,
partition ord4 values less than (to_date('1992-04-
01','YYYY-MM-DD'))
tablespace ts_o4
,
partition ord5 values less than (to_date('1992-05-
01','YYYY-MM-DD'))
tablespace ts_o5
,
partition ord6 values less than (to_date('1992-06-
01','YYYY-MM-DD'))
tablespace ts_o6
,
partition ord7 values less than (to_date('1992-07-
01','YYYY-MM-DD'))
tablespace ts_o7
,
partition ord8 values less than (to_date('1992-08-
01','YYYY-MM-DD'))
tablespace ts_o8
,
partition ord9 values less than (to_date('1992-09-
01','YYYY-MM-DD'))
tablespace ts_o9
,
partition ord10 values less than (to_date('1992-10-
01','YYYY-MM-DD'))
tablespace ts_o10
,
partition ord11 values less than (to_date('1992-11-
01','YYYY-MM-DD'))
tablespace ts_o11
,
partition ord12 values less than (to_date('1992-12-
01','YYYY-MM-DD'))
tablespace ts_o12
,
partition ord13 values less than (to_date('1993-01-
01','YYYY-MM-DD'))
tablespace ts_o13
,
partition ord14 values less than (to_date('1993-02-
01','YYYY-MM-DD'))
tablespace ts_o14
,
partition ord15 values less than (to_date('1993-03-
01','YYYY-MM-DD'))
tablespace ts_o15
,
partition ord16 values less than (to_date('1993-04-
01','YYYY-MM-DD'))
tablespace ts_o16
,
partition ord17 values less than (to_date('1993-05-
01','YYYY-MM-DD'))
tablespace ts_o17
,
partition ord18 values less than (to_date('1993-06-
01','YYYY-MM-DD'))
tablespace ts_o18
,
partition ord19 values less than (to_date('1993-07-
01','YYYY-MM-DD'))
tablespace ts_o19
,
partition ord20 values less than (to_date('1993-08-
```

```
01','YYYY-MM-DD'))
tablespace ts_o20
,
partition ord21 values less than (to_date('1993-09-
01','YYYY-MM-DD'))
tablespace ts_o21
,
partition ord22 values less than (to_date('1993-10-
01','YYYY-MM-DD'))
tablespace ts_o22
,
partition ord23 values less than (to_date('1993-11-
01','YYYY-MM-DD'))
tablespace ts_o23
,
partition ord24 values less than (to_date('1993-12-
01','YYYY-MM-DD'))
tablespace ts_o24
,
partition ord25 values less than (to_date('1994-01-
01','YYYY-MM-DD'))
tablespace ts_o25
,
partition ord26 values less than (to_date('1994-02-
01','YYYY-MM-DD'))
tablespace ts_o26
,
partition ord27 values less than (to_date('1994-03-
01','YYYY-MM-DD'))
tablespace ts_o27
,
partition ord28 values less than (to_date('1994-04-
01','YYYY-MM-DD'))
tablespace ts_o28
,
partition ord29 values less than (to_date('1994-05-
01','YYYY-MM-DD'))
tablespace ts_o29
,
partition ord30 values less than (to_date('1994-06-
01','YYYY-MM-DD'))
tablespace ts_o30
,
partition ord31 values less than (to_date('1994-07-
01','YYYY-MM-DD'))
tablespace ts_o31
,
partition ord32 values less than (to_date('1994-08-
01','YYYY-MM-DD'))
tablespace ts_o32
,
partition ord33 values less than (to_date('1994-09-
01','YYYY-MM-DD'))
tablespace ts_o33
,
partition ord34 values less than (to_date('1994-10-
01','YYYY-MM-DD'))
tablespace ts_o34
,
partition ord35 values less than (to_date('1994-11-
01','YYYY-MM-DD'))
tablespace ts_o35
,
partition ord36 values less than (to_date('1994-12-
01','YYYY-MM-DD'))
tablespace ts_o36
,
partition ord37 values less than (to_date('1995-01-
01','YYYY-MM-DD'))
tablespace ts_o37
,
partition ord38 values less than (to_date('1995-02-
01','YYYY-MM-DD'))
tablespace ts_o38
,
partition ord39 values less than (to_date('1995-03-
01','YYYY-MM-DD'))
tablespace ts_o39
,
partition ord40 values less than (to_date('1995-04-
01','YYYY-MM-DD'))
tablespace ts_o40
,
partition ord41 values less than (to_date('1995-05-
01','YYYY-MM-DD'))
tablespace ts_o41
,
partition ord42 values less than (to_date('1995-06-
01','YYYY-MM-DD'))
tablespace ts_o42
,
partition ord43 values less than (to_date('1995-07-
01','YYYY-MM-DD'))
tablespace ts_o43
,
partition ord44 values less than (to_date('1995-08-
01','YYYY-MM-DD'))
tablespace ts_o44
,
partition ord45 values less than (to_date('1995-09-
01','YYYY-MM-DD'))
tablespace ts_o45
,
partition ord46 values less than (to_date('1995-10-
01','YYYY-MM-DD'))
tablespace ts_o46
,
partition ord47 values less than (to_date('1995-11-
01','YYYY-MM-DD'))
tablespace ts_o47
,
partition ord48 values less than (to_date('1995-12-
01','YYYY-MM-DD'))
tablespace ts_o48
,
partition ord49 values less than (to_date('1996-01-
01','YYYY-MM-DD'))
tablespace ts_o49
,
partition ord50 values less than (to_date('1996-02-
01','YYYY-MM-DD'))
tablespace ts_o50
,
partition ord51 values less than (to_date('1996-03-
01','YYYY-MM-DD'))
tablespace ts_o51
,
partition ord52 values less than (to_date('1996-04-
01','YYYY-MM-DD'))
tablespace ts_o52
,
partition ord53 values less than (to_date('1996-05-
01','YYYY-MM-DD'))
tablespace ts_o53
,
partition ord54 values less than (to_date('1996-06-
01','YYYY-MM-DD'))
tablespace ts_o54
,
partition ord55 values less than (to_date('1996-07-
01','YYYY-MM-DD'))
tablespace ts_o55
,
partition ord56 values less than (to_date('1996-08-
01','YYYY-MM-DD'))
tablespace ts_o56
,
partition ord57 values less than (to_date('1996-09-
01','YYYY-MM-DD'))
tablespace ts_o57
,
partition ord58 values less than (to_date('1996-10-
01','YYYY-MM-DD'))
tablespace ts_o58
,
partition ord59 values less than (to_date('1996-11-
01','YYYY-MM-DD'))
tablespace ts_o59
,
partition ord60 values less than (to_date('1996-12-
01','YYYY-MM-DD'))
tablespace ts_o60
,
partition ord61 values less than (to_date('1997-01-
01','YYYY-MM-DD'))
tablespace ts_o61
,
partition ord62 values less than (to_date('1997-02-
01','YYYY-MM-DD'))
tablespace ts_o62
,
partition ord63 values less than (to_date('1997-03-
01','YYYY-MM-DD'))
tablespace ts_o63
,
partition ord64 values less than (to_date('1997-04-
01','YYYY-MM-DD'))
tablespace ts_o64
,
partition ord65 values less than (to_date('1997-05-
```

```
01','YYYY-MM-DD'))
tablespace ts_o65
,
partition ord66 values less than (to_date('1997-06-
01','YYYY-MM-DD'))
tablespace ts_o66
,
partition ord67 values less than (to_date('1997-07-
01','YYYY-MM-DD'))
tablespace ts_o67
,
partition ord68 values less than (to_date('1997-08-
01','YYYY-MM-DD'))
tablespace ts_o68
,
partition ord69 values less than (to_date('1997-09-
01','YYYY-MM-DD'))
tablespace ts_o69
,
partition ord70 values less than (to_date('1997-10-
01','YYYY-MM-DD'))
tablespace ts_o70
,
partition ord71 values less than (to_date('1997-11-
01','YYYY-MM-DD'))
tablespace ts_o71
,
partition ord72 values less than (to_date('1997-12-
01','YYYY-MM-DD'))
tablespace ts_o72
,
partition ord73 values less than (to_date('1998-01-
01','YYYY-MM-DD'))
tablespace ts_o73
,
partition ord74 values less than (to_date('1998-02-
01','YYYY-MM-DD'))
tablespace ts_o74
,
partition ord75 values less than (to_date('1998-03-
01','YYYY-MM-DD'))
tablespace ts_o75
,
partition ord76 values less than (to_date('1998-04-
01','YYYY-MM-DD'))
tablespace ts_o76
,
partition ord77 values less than (to_date('1998-05-
01','YYYY-MM-DD'))
tablespace ts_o77
,
partition ord78 values less than (to_date('1998-06-
01','YYYY-MM-DD'))
tablespace ts_o78
,
partition ord79 values less than (to_date('1998-07-
01','YYYY-MM-DD'))
tablespace ts_o79
,
partition ord80 values less than (to_date('1998-08-
01','YYYY-MM-DD'))
tablespace ts_o80
,
partition ord81 values less than (to_date('1998-09-
01','YYYY-MM-DD'))
tablespace ts_o81
,
partition ord82 values less than (to_date('1998-10-
01','YYYY-MM-DD'))
tablespace ts_o82
,
partition ord83 values less than (to_date('1998-11-
01','YYYY-MM-DD'))
tablespace ts_o83
,
partition ord84 values less than (MAXVALUE)
tablespace ts_o84
)
;
drop table partsupp;
create table partsupp(
    ps_partkey          number NOT NULL,
    ps_suppkey          number NOT NULL,
    ps_supplycost       number NOT NULL,
    ps_availqty         number ,
    ps_comment          varchar(199)
)
pctfree 0
pctused 99
tablespace ts_ps
storage (freelists 99 freelist groups 2)
parallel
partition by hash (ps_suppkey)
partitions 16
;
drop table part;
create table part(
    p_partkey           number NOT NULL,
    p_type              varchar(25) ,
    p_size              number ,
    p_brand             char(10) ,
    p_name              varchar(55) ,
    p_container         char(10) ,
    p_mfgr              char(25) ,
    p_retailprice       number ,
    p_comment           varchar(23)
)
pctfree 0
pctused 99
tablespace ts_p
storage (freelists 99 freelist groups 2)
parallel
partition by hash (p_partkey)
partitions 16
;
drop table customer;
create table customer(
    c_custkey           number NOT NULL,
    c_mktsegment        char(10) ,
    c_nationkey         number ,
    c_name              varchar(25) ,
    c_address           varchar(40) ,
    c_phone             char(15) ,
    c_acctbal           number ,
    c_comment           varchar(117)
)
pctfree 0
pctused 99
tablespace ts_c
storage (freelists 99 freelist groups 2)
parallel
partition by hash (c_custkey)
partitions 16
;
drop table supplier;
create table supplier(
    s_suppkey           number NOT NULL,
    s_nationkey         number ,
    s_comment           varchar(101) ,
    s_name              char(25) ,
    s_address           varchar(40) ,
    s_phone             char(15) ,
    s_acctbal           number
)
pctfree 0
pctused 99
tablespace ts_s
storage (freelists 99 freelist groups 2)
parallel
;
drop table nation;
create table nation(
    n_nationkey         number NOT NULL,
    n_name              char(25) ,
    n_regionkey         number ,
    n_comment           varchar(152)
)
tablespace ts_default
;
drop table region;
create table region(
    r_regionkey         number ,
    r_name              char(25) ,
    r_comment           varchar(152)
)
tablespace ts_default
;
}
*wait
# altering tpcd's temp tablespace
*sql
{
alter user tpcd temporary tablespace ts_temp;
}
*wait
# altering tpcd's default tablespace
*sql
{
alter user tpcd default tablespace ts_default;
}
```

```
*wait
*sql
{
connect tpcd/tpcd
@?/rdbms/admin/utlxplan.sql;
}
*wait
*wait
*bgoff
%e-scuto

=====================================================
```

3tera_li_dapop.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load1
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load2
\sqlldr {}

*load3
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dapop

*bgon=168
#########################################################
###################
# Database Population Phase
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem1.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_1_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem85.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_85_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem2.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_2_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem86.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_86_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem3.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_3_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem87.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_87_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem4.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_4_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem88.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_88_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem5.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_5_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem89.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_89_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem6.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_6_1.lo
```

```
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem90.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_90_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem7.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_7_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem91.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_91_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem8.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_8_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem92.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_92_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem9.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_9_1.lo
g direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem93.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_93_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem10.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_10_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem94.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_94_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem11.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_11_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem95.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_95_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem12.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_12_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem96.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_96_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem13.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_13_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem97.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_97_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem14.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_14_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem98.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_98_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem15.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_15_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem99.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_99_43.
log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem16.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_16_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem100.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_100_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem17.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_17_1.l
og direct=true parallel=true
}
*load2
{
```

```
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem101.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_101_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem18.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_18_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem102.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_102_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem19.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_19_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem103.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_103_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem20.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_20_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem104.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_104_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem21.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_21_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem105.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_105_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem22.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_22_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem106.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_106_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem23.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_23_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem107.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_107_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem24.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_24_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem108.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_108_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem25.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_25_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem109.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_109_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem26.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_26_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem110.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_110_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem27.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_27_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem111.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_111_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem28.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_28_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem112.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_112_43
```

```
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem29.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_29_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem113.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_113_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem30.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_30_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem114.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_114_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem31.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_31_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem115.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_115_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem32.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_32_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem116.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_116_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem33.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_33_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem117.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_117_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem34.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_34_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem118.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_118_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem35.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_35_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem119.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_119_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem36.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_36_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem120.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_120_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem37.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_37_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem121.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_121_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem38.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_38_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem122.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_122_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem39.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_39_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem123.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_123_43
.log direct=true parallel=true
}
*load1
{
```

```
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem40.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_40_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem124.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_124_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem41.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_41_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem125.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_125_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem42.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_42_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem126.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_126_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem43.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_43_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem127.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_127_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem44.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_44_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem128.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_128_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem45.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_45_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem129.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_129_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem46.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_46_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem130.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_130_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem47.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_47_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem131.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_131_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem48.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_48_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem132.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_132_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem49.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_49_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem133.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_133_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem50.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_50_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem134.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_134_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem51.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_51_1.l
```

```
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem135.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_135_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem52.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_52_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem136.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_136_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem53.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_53_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem137.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_137_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem54.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_54_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem138.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_138_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem55.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_55_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem139.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_139_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem56.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_56_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem140.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_140_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem57.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_57_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem141.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_141_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem58.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_58_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem142.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_142_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem59.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_59_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem143.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_143_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem60.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_60_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem144.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_144_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem61.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_61_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem145.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_145_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem62.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_62_1.l
og direct=true parallel=true
}
*load2
{
```

```
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem146.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_146_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem63.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_63_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem147.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_147_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem64.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_64_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem148.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_148_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem65.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_65_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem149.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_149_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem66.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_66_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem150.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_150_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem67.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_67_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem151.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_151_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem68.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_68_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem152.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_152_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem69.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_69_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem153.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_153_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem70.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_70_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem154.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_154_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem71.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_71_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem155.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_155_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem72.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_72_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem156.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_156_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem73.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_73_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem157.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_157_43
```

```
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem74.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_74_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem158.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_158_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem75.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_75_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem159.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_159_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem76.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_76_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem160.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_160_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem77.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_77_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem161.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_161_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem78.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_78_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem162.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_162_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem79.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_79_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem163.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_163_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem80.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_80_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem164.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_164_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem81.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_81_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem165.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_165_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem82.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_82_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem166.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_166_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem83.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_83_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem167.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_167_43
.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem84.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_84_1.l
og direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/linei
tem168.ctl log=
/export/home/oracle/kit/load/other/3tb/lineitem_168_43
.log direct=true parallel=true
}
*wait
*wait
```

```
*wait
*wait
*bgoff
%e-dapop

================================================
```

3tera_or_dapop.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load1
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load2
\sqlldr {}

*load3
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dapop
*bgon=168
##################################################
######################
# Database Population Phase
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s1.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_1_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s85.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_85_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s2.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_2_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s86.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_86_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s3.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_3_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s87.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_87_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s4.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_4_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s88.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_88_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s5.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_5_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s89.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_89_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s6.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_6_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
```

```
s90.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_90_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s7.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_7_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s91.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_91_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s8.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_8_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s92.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_92_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s9.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_9_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s93.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_93_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s10.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_10_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s94.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_94_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s11.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_11_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s95.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_95_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s12.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_12_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s96.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_96_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s13.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_13_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s97.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_97_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s14.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_14_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s98.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_98_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s15.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_15_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s99.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_99_43.lo
g direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s16.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_16_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s100.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_100_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s17.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_17_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s101.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_101_43.l
og direct=true parallel=true
}
```

```
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders18.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_18_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders102.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_102_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders19.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_19_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders103.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_103_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders20.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_20_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders104.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_104_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders21.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_21_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders105.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_105_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders22.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_22_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders106.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_106_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders23.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_23_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders107.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_107_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders24.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_24_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders108.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_108_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders25.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_25_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders109.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_109_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders26.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_26_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders110.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_110_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders27.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_27_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders111.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_111_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders28.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_28_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders112.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_112_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
```

```
s29.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_29_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s113.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_113_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s30.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_30_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s114.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_114_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s31.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_31_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s115.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_115_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s32.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_32_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s116.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_116_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s33.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_33_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s117.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_117_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s34.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_34_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s118.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_118_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s35.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_35_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s119.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_119_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s36.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_36_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s120.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_120_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s37.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_37_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s121.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_121_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s38.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_38_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s122.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_122_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s39.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_39_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s123.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_123_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s40.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_40_1.log
direct=true parallel=true
}
```

```
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders124.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_124_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders41.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_41_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders125.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_125_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders42.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_42_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders126.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_126_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders43.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_43_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders127.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_127_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders44.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_44_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders128.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_128_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders45.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_45_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders129.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_129_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders46.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_46_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders130.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_130_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders47.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_47_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders131.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_131_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders48.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_48_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders132.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_132_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders49.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_49_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders133.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_133_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders50.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_50_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders134.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_134_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders51.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_51_1.log direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
```

```
s135.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_135_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s52.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_52_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s136.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_136_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s53.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_53_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s137.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_137_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s54.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_54_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s138.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_138_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s55.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_55_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s139.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_139_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s56.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_56_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s140.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_140_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s57.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_57_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s141.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_141_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s58.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_58_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s142.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_142_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s59.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_59_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s143.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_143_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s60.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_60_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s144.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_144_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s61.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_61_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s145.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_145_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s62.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_62_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s146.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_146_43.l
og direct=true parallel=true
}
```

```
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders63.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_63_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders147.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_147_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders64.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_64_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders148.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_148_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders65.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_65_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders149.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_149_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders66.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_66_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders150.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_150_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders67.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_67_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders151.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_151_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders68.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_68_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders152.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_152_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders69.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_69_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders153.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_153_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders70.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_70_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders154.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_154_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders71.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_71_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders155.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_155_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders72.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_72_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders156.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_156_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders73.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_73_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/orders157.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_157_43.log direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
```

```
s74.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_74_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s158.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_158_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s75.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_75_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s159.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_159_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s76.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_76_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s160.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_160_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s77.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_77_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s161.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_161_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s78.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_78_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s162.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_162_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s79.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_79_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s163.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_163_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s80.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_80_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s164.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_164_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s81.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_81_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s165.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_165_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s82.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_82_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s166.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_166_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s83.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_83_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s167.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_167_43.l
og direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s84.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_84_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/order
s168.ctl log=
/export/home/oracle/kit/load/other/3tb/orders_168_43.l
og direct=true parallel=true
}
*wait
*wait
*wait
*wait
*bgoff
%e-dapop
```

```
=====================================================
```

3tera_papsupcsnr_dapop.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load1
\sqlldr {}

*load2
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dapop
*bgon=300
#####################################################
######################
# Database Population Phase
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp1.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_1.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp33.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_33.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp2.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_2.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp34.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_34.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp3.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_3.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp35.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_35.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp4.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_4.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp36.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_36.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp5.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_5.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp37.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_37.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp6.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_6.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/parts
upp38.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_38.log
```

```
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp7.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_7.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp39.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_39.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp8.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_8.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp40.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_40.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp9.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_9.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp41.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_41.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp10.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_10.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp42.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_42.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp11.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_11.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp43.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_43.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp12.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_12.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp44.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_44.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp13.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_13.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp45.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_45.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp14.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_14.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp46.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_46.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp15.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_15.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp47.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_47.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp16.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_16.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp48.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_48.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp17.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_17.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp49.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_49.log
direct=true parallel=true
}
*load1
{
```

```
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp18.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_18.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp50.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_50.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp19.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_19.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp51.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_51.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp20.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_20.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp52.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_52.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp21.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_21.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp53.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_53.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp22.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_22.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp54.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_54.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp23.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_23.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp55.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_55.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp24.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_24.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp56.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_56.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp25.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_25.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp57.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_57.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp26.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_26.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp58.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_58.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp27.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_27.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp59.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_59.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp28.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_28.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp60.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_60.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp29.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_29.log
```

```
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp61.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_61.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp30.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_30.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp62.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_62.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp31.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_31.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp63.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_63.log
direct=true parallel=true
}
*load1
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp32.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_32.log
direct=true parallel=true
}
*load2
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/partsupp64.ctl log=
/export/home/oracle/kit/load/other/3tb/partsupp_64.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_1.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part33.ctl log=
/export/home/oracle/kit/load/other/3tb/part_33.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_2.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part34.ctl log=
/export/home/oracle/kit/load/other/3tb/part_34.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part3.ctl log=
/export/home/oracle/kit/load/other/3tb/part_3.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part35.ctl log=
/export/home/oracle/kit/load/other/3tb/part_35.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4.ctl log=
/export/home/oracle/kit/load/other/3tb/part_4.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part36.ctl log=
/export/home/oracle/kit/load/other/3tb/part_36.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5.ctl log=
/export/home/oracle/kit/load/other/3tb/part_5.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part37.ctl log=
/export/home/oracle/kit/load/other/3tb/part_37.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part6.ctl log=
/export/home/oracle/kit/load/other/3tb/part_6.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part38.ctl log=
/export/home/oracle/kit/load/other/3tb/part_38.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part7.ctl log=
/export/home/oracle/kit/load/other/3tb/part_7.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part39.ctl log=
/export/home/oracle/kit/load/other/3tb/part_39.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part8.ctl log=
/export/home/oracle/kit/load/other/3tb/part_8.log
direct=true parallel=true
}
*load3
{
```

```
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
0.ctl log=
/export/home/oracle/kit/load/other/3tb/part_40.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part9
.ctl log=
/export/home/oracle/kit/load/other/3tb/part_9.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_41.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
0.ctl log=
/export/home/oracle/kit/load/other/3tb/part_10.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_42.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_11.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
3.ctl log=
/export/home/oracle/kit/load/other/3tb/part_43.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_12.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
4.ctl log=
/export/home/oracle/kit/load/other/3tb/part_44.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
3.ctl log=
/export/home/oracle/kit/load/other/3tb/part_13.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
5.ctl log=
/export/home/oracle/kit/load/other/3tb/part_45.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
4.ctl log=
/export/home/oracle/kit/load/other/3tb/part_14.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
6.ctl log=
/export/home/oracle/kit/load/other/3tb/part_46.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
5.ctl log=
/export/home/oracle/kit/load/other/3tb/part_15.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
7.ctl log=
/export/home/oracle/kit/load/other/3tb/part_47.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
6.ctl log=
/export/home/oracle/kit/load/other/3tb/part_16.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
8.ctl log=
/export/home/oracle/kit/load/other/3tb/part_48.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
7.ctl log=
/export/home/oracle/kit/load/other/3tb/part_17.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part4
9.ctl log=
/export/home/oracle/kit/load/other/3tb/part_49.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
8.ctl log=
/export/home/oracle/kit/load/other/3tb/part_18.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
0.ctl log=
/export/home/oracle/kit/load/other/3tb/part_50.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part1
9.ctl log=
/export/home/oracle/kit/load/other/3tb/part_19.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_51.log
```

```
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
0.ctl log=
/export/home/oracle/kit/load/other/3tb/part_20.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_52.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_21.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
3.ctl log=
/export/home/oracle/kit/load/other/3tb/part_53.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_22.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
4.ctl log=
/export/home/oracle/kit/load/other/3tb/part_54.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
3.ctl log=
/export/home/oracle/kit/load/other/3tb/part_23.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
5.ctl log=
/export/home/oracle/kit/load/other/3tb/part_55.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
4.ctl log=
/export/home/oracle/kit/load/other/3tb/part_24.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
6.ctl log=
/export/home/oracle/kit/load/other/3tb/part_56.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
5.ctl log=
/export/home/oracle/kit/load/other/3tb/part_25.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
7.ctl log=
/export/home/oracle/kit/load/other/3tb/part_57.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
6.ctl log=
/export/home/oracle/kit/load/other/3tb/part_26.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
8.ctl log=
/export/home/oracle/kit/load/other/3tb/part_58.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
7.ctl log=
/export/home/oracle/kit/load/other/3tb/part_27.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part5
9.ctl log=
/export/home/oracle/kit/load/other/3tb/part_59.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
8.ctl log=
/export/home/oracle/kit/load/other/3tb/part_28.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part6
0.ctl log=
/export/home/oracle/kit/load/other/3tb/part_60.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part2
9.ctl log=
/export/home/oracle/kit/load/other/3tb/part_29.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part6
1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_61.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part3
0.ctl log=
/export/home/oracle/kit/load/other/3tb/part_30.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part6
2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_62.log
direct=true parallel=true
}
*load3
{
```

```
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part3
1.ctl log=
/export/home/oracle/kit/load/other/3tb/part_31.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part6
3.ctl log=
/export/home/oracle/kit/load/other/3tb/part_63.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part3
2.ctl log=
/export/home/oracle/kit/load/other/3tb/part_32.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/part6
4.ctl log=
/export/home/oracle/kit/load/other/3tb/part_64.log
direct=true parallel=true
}
*wait
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer1.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_1.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer33.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_33.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer2.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_2.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer34.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_34.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer3.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_3.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer35.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_35.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer4.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_4.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer36.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_36.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer5.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_5.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer37.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_37.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer6.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_6.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer38.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_38.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer7.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_7.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer39.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_39.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer8.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_8.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer40.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_40.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer9.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_9.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer41.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_41.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer10.ctl log=
```

```
/export/home/oracle/kit/load/other/3tb/customer_10.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer42.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_42.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer11.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_11.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer43.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_43.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer12.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_12.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer44.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_44.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer13.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_13.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer45.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_45.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer14.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_14.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer46.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_46.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer15.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_15.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer47.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_47.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer16.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_16.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer48.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_48.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer17.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_17.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer49.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_49.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer18.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_18.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer50.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_50.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer19.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_19.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer51.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_51.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer20.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_20.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer52.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_52.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer21.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_21.log
direct=true parallel=true
}
*load3
```

```
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer53.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_53.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer22.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_22.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer54.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_54.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer23.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_23.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer55.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_55.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer24.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_24.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer56.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_56.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer25.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_25.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer57.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_57.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer26.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_26.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer58.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_58.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer27.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_27.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer59.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_59.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer28.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_28.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer60.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_60.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer29.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_29.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer61.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_61.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer30.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_30.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer62.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_62.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer31.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_31.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer63.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_63.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer32.ctl log=
/export/home/oracle/kit/load/other/3tb/customer_32.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/custo
mer64.ctl log=
```

```
/export/home/oracle/kit/load/other/3tb/customer_64.log
direct=true parallel=true
}
*wait
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/natio
n.ctl log=
/export/home/oracle/kit/load/other/3tb/nation.log
direct=true
}
*wait
*wait
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/regio
n.ctl log=
/export/home/oracle/kit/load/other/3tb/region.log
direct=true
}
*wait
*wait
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier1.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_1.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier33.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_33.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier2.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_2.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier34.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_34.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier3.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_3.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier35.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_35.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier4.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_4.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier36.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_36.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier5.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_5.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier37.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_37.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier6.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_6.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier38.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_38.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier7.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_7.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier39.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_39.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier8.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_8.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier40.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_40.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier9.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_9.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier41.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_41.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier10.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_10.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
```

```
ier42.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_42.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier11.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_11.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier43.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_43.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier12.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_12.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier44.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_44.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier13.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_13.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier45.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_45.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier14.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_14.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier46.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_46.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier15.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_15.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier47.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_47.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier16.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_16.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier48.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_48.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier17.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_17.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier49.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_49.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier18.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_18.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier50.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_50.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier19.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_19.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier51.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_51.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier20.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_20.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier52.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_52.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier21.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_21.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/suppl
ier53.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_53.log
direct=true parallel=true
}
```

```
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier22.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_22.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier54.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_54.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier23.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_23.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier55.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_55.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier24.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_24.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier56.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_56.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier25.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_25.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier57.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_57.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier26.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_26.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier58.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_58.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier27.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_27.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier59.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_59.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier28.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_28.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier60.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_60.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier29.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_29.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier61.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_61.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier30.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_30.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier62.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_62.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier31.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_31.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier63.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_63.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier32.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_32.log
direct=true parallel=true
}
*load3
{
tpcd/tpcd
control=/export/home/oracle/kit/load/control/3tb/supplier64.ctl log=
/export/home/oracle/kit/load/other/3tb/supplier_64.log
direct=true parallel=true
}
*wait
*wait
*wait
*bgoff
```

```
%e-dapop
*wait
*wait
*wait
*bgoff
%e-dapop
=====================================================
=====================================================
```

3tera_ixcre.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load1
\sqlldr {}

*load2
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-ixcre
*bgon=1
#####################################################
######################
# Index Creation Phase

*sql
{
connect tpcd/tpcd;
drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_i_lorderkey
storage (freelists 99 freelist groups 2)
parallel
;
}
*sql
{
connect tpcd/tpcd;
drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_i_oorderkey
storage (freelists 99 freelist groups 2)
parallel
global partition by range (o_orderkey)
(
partition o_ordk1 values less than (4394531),
partition o_ordk2 values less than (8789062),
partition o_ordk3 values less than (13183594),
partition o_ordk4 values less than (17578125),
partition o_ordk5 values less than (21972656),
partition o_ordk6 values less than (26367187),
partition o_ordk7 values less than (30761719),
partition o_ordk8 values less than (35156250),
partition o_ordk9 values less than (39550781),
partition o_ordk10 values less than (43945312),
partition o_ordk11 values less than (48339844),
partition o_ordk12 values less than (52734375),
partition o_ordk13 values less than (57128906),
partition o_ordk14 values less than (61523437),
partition o_ordk15 values less than (65917969),
partition o_ordk16 values less than (70312500),
partition o_ordk17 values less than (74707031),
partition o_ordk18 values less than (79101562),
partition o_ordk19 values less than (83496094),
partition o_ordk20 values less than (87890625),
partition o_ordk21 values less than (92285156),
partition o_ordk22 values less than (96679687),
partition o_ordk23 values less than (101074219),
partition o_ordk24 values less than (105468750),
partition o_ordk25 values less than (109863281),
partition o_ordk26 values less than (114257812),
partition o_ordk27 values less than (118652344),
partition o_ordk28 values less than (123046875),
partition o_ordk29 values less than (127441406),
partition o_ordk30 values less than (131835937),
partition o_ordk31 values less than (136230469),
partition o_ordk32 values less than (140625000),
partition o_ordk33 values less than (145019531),
partition o_ordk34 values less than (149414062),
partition o_ordk35 values less than (153808594),
partition o_ordk36 values less than (158203125),
partition o_ordk37 values less than (162597656),
partition o_ordk38 values less than (166992187),
partition o_ordk39 values less than (171386719),
partition o_ordk40 values less than (175781250),
partition o_ordk41 values less than (180175781),
partition o_ordk42 values less than (184570312),
partition o_ordk43 values less than (188964844),
partition o_ordk44 values less than (193359375),
partition o_ordk45 values less than (197753906),
partition o_ordk46 values less than (202148437),
partition o_ordk47 values less than (206542969),
partition o_ordk48 values less than (210937500),
partition o_ordk49 values less than (215332031),
partition o_ordk50 values less than (219726562),
partition o_ordk51 values less than (224121094),
partition o_ordk52 values less than (228515625),
partition o_ordk53 values less than (232910156),
partition o_ordk54 values less than (237304687),
partition o_ordk55 values less than (241699219),
partition o_ordk56 values less than (246093750),
partition o_ordk57 values less than (250488281),
partition o_ordk58 values less than (254882812),
partition o_ordk59 values less than (259277344),
partition o_ordk60 values less than (263671875),
partition o_ordk61 values less than (268066406),
partition o_ordk62 values less than (272460937),
```

```
partition o_ordk63 values less than (276855469),
partition o_ordk64 values less than (281250000),
partition o_ordk65 values less than (285644531),
partition o_ordk66 values less than (290039062),
partition o_ordk67 values less than (294433594),
partition o_ordk68 values less than (298828125),
partition o_ordk69 values less than (303222656),
partition o_ordk70 values less than (307617187),
partition o_ordk71 values less than (312011719),
partition o_ordk72 values less than (316406250),
partition o_ordk73 values less than (320800781),
partition o_ordk74 values less than (325195312),
partition o_ordk75 values less than (329589844),
partition o_ordk76 values less than (333984375),
partition o_ordk77 values less than (338378906),
partition o_ordk78 values less than (342773437),
partition o_ordk79 values less than (347167969),
partition o_ordk80 values less than (351562500),
partition o_ordk81 values less than (355957031),
partition o_ordk82 values less than (360351562),
partition o_ordk83 values less than (364746094),
partition o_ordk84 values less than (369140625),
partition o_ordk85 values less than (373535156),
partition o_ordk86 values less than (377929687),
partition o_ordk87 values less than (382324219),
partition o_ordk88 values less than (386718750),
partition o_ordk89 values less than (391113281),
partition o_ordk90 values less than (395507812),
partition o_ordk91 values less than (399902344),
partition o_ordk92 values less than (404296875),
partition o_ordk93 values less than (408691406),
partition o_ordk94 values less than (413085937),
partition o_ordk95 values less than (417480469),
partition o_ordk96 values less than (421875000),
partition o_ordk97 values less than (426269531),
partition o_ordk98 values less than (430664062),
partition o_ordk99 values less than (435058594),
partition o_ordk100 values less than (439453125),
partition o_ordk101 values less than (443847656),
partition o_ordk102 values less than (448242187),
partition o_ordk103 values less than (452636719),
partition o_ordk104 values less than (457031250),
partition o_ordk105 values less than (461425781),
partition o_ordk106 values less than (465820312),
partition o_ordk107 values less than (470214844),
partition o_ordk108 values less than (474609375),
partition o_ordk109 values less than (479003906),
partition o_ordk110 values less than (483398437),
partition o_ordk111 values less than (487792969),
partition o_ordk112 values less than (492187500),
partition o_ordk113 values less than (496582031),
partition o_ordk114 values less than (500976562),
partition o_ordk115 values less than (505371093),
partition o_ordk116 values less than (509765625),
partition o_ordk117 values less than (514160156),
partition o_ordk118 values less than (518554687),
partition o_ordk119 values less than (522949218),
partition o_ordk120 values less than (527343750),
partition o_ordk121 values less than (531738281),
partition o_ordk122 values less than (536132812),
partition o_ordk123 values less than (540527343),
partition o_ordk124 values less than (544921875),
partition o_ordk125 values less than (549316406),
partition o_ordk126 values less than (553710937),
partition o_ordk127 values less than (558105468),
partition o_ordk128 values less than (562500000),
partition o_ordk129 values less than (566894531),
partition o_ordk130 values less than (571289062),
partition o_ordk131 values less than (575683593),
partition o_ordk132 values less than (580078125),
partition o_ordk133 values less than (584472656),
partition o_ordk134 values less than (588867187),
partition o_ordk135 values less than (593261718),
partition o_ordk136 values less than (597656250),
partition o_ordk137 values less than (602050781),
partition o_ordk138 values less than (606445312),
partition o_ordk139 values less than (610839843),
partition o_ordk140 values less than (615234375),
partition o_ordk141 values less than (619628906),
partition o_ordk142 values less than (624023437),
partition o_ordk143 values less than (628417968),
partition o_ordk144 values less than (632812500),
partition o_ordk145 values less than (637207031),
partition o_ordk146 values less than (641601562),
partition o_ordk147 values less than (645996093),
partition o_ordk148 values less than (650390625),
partition o_ordk149 values less than (654785156),
partition o_ordk150 values less than (659179687),
partition o_ordk151 values less than (663574218),
partition o_ordk152 values less than (667968750),
partition o_ordk153 values less than (672363281),
partition o_ordk154 values less than (676757812),
partition o_ordk155 values less than (681152343),
partition o_ordk156 values less than (685546875),
partition o_ordk157 values less than (689941406),
partition o_ordk158 values less than (694335937),
partition o_ordk159 values less than (698730468),
partition o_ordk160 values less than (703125000),
partition o_ordk161 values less than (707519531),
partition o_ordk162 values less than (711914062),
partition o_ordk163 values less than (716308593),
partition o_ordk164 values less than (720703125),
partition o_ordk165 values less than (725097656),
partition o_ordk166 values less than (729492187),
partition o_ordk167 values less than (733886718),
partition o_ordk168 values less than (738281250),
partition o_ordk169 values less than (742675781),
partition o_ordk170 values less than (747070312),
partition o_ordk171 values less than (751464843),
partition o_ordk172 values less than (755859375),
partition o_ordk173 values less than (760253906),
partition o_ordk174 values less than (764648437),
partition o_ordk175 values less than (769042968),
partition o_ordk176 values less than (773437500),
partition o_ordk177 values less than (777832031),
partition o_ordk178 values less than (782226562),
partition o_ordk179 values less than (786621093),
partition o_ordk180 values less than (791015625),
partition o_ordk181 values less than (795410156),
partition o_ordk182 values less than (799804687),
partition o_ordk183 values less than (804199218),
partition o_ordk184 values less than (808593750),
partition o_ordk185 values less than (812988281),
partition o_ordk186 values less than (817382812),
partition o_ordk187 values less than (821777343),
partition o_ordk188 values less than (826171875),
partition o_ordk189 values less than (830566406),
partition o_ordk190 values less than (834960937),
partition o_ordk191 values less than (839355468),
partition o_ordk192 values less than (843750000),
partition o_ordk193 values less than (848144531),
partition o_ordk194 values less than (852539062),
partition o_ordk195 values less than (856933593),
partition o_ordk196 values less than (861328125),
partition o_ordk197 values less than (865722656),
partition o_ordk198 values less than (870117187),
partition o_ordk199 values less than (874511718),
partition o_ordk200 values less than (878906250),
partition o_ordk201 values less than (883300781),
partition o_ordk202 values less than (887695312),
partition o_ordk203 values less than (892089843),
partition o_ordk204 values less than (896484375),
partition o_ordk205 values less than (900878906),
partition o_ordk206 values less than (905273437),
partition o_ordk207 values less than (909667968),
partition o_ordk208 values less than (914062500),
partition o_ordk209 values less than (918457031),
partition o_ordk210 values less than (922851562),
partition o_ordk211 values less than (927246093),
partition o_ordk212 values less than (931640625),
partition o_ordk213 values less than (936035156),
partition o_ordk214 values less than (940429687),
partition o_ordk215 values less than (944824218),
partition o_ordk216 values less than (949218750),
partition o_ordk217 values less than (953613281),
partition o_ordk218 values less than (958007812),
partition o_ordk219 values less than (962402343),
partition o_ordk220 values less than (966796875),
partition o_ordk221 values less than (971191406),
partition o_ordk222 values less than (975585937),
partition o_ordk223 values less than (979980468),
partition o_ordk224 values less than (984375000),
partition o_ordk225 values less than (988769531),
partition o_ordk226 values less than (993164062),
partition o_ordk227 values less than (997558593),
partition o_ordk228 values less than (1001953124),
partition o_ordk229 values less than (1006347656),
partition o_ordk230 values less than (1010742187),
partition o_ordk231 values less than (1015136718),
partition o_ordk232 values less than (1019531249),
partition o_ordk233 values less than (1023925781),
partition o_ordk234 values less than (1028320312),
partition o_ordk235 values less than (1032714843),
partition o_ordk236 values less than (1037109374),
partition o_ordk237 values less than (1041503906),
partition o_ordk238 values less than (1045898437),
partition o_ordk239 values less than (1050292968),
partition o_ordk240 values less than (1054687499),
partition o_ordk241 values less than (1059082031),
partition o_ordk242 values less than (1063476562),
```

```
partition o_ordk243 values less than (1067871093),
partition o_ordk244 values less than (1072265624),
partition o_ordk245 values less than (1076660156),
partition o_ordk246 values less than (1081054687),
partition o_ordk247 values less than (1085449218),
partition o_ordk248 values less than (1089843749),
partition o_ordk249 values less than (1094238281),
partition o_ordk250 values less than (1098632812),
partition o_ordk251 values less than (1103027343),
partition o_ordk252 values less than (1107421874),
partition o_ordk253 values less than (1111816406),
partition o_ordk254 values less than (1116210937),
partition o_ordk255 values less than (1120605468),
partition o_ordk256 values less than (1124999999),
partition o_ordk257 values less than (1129394531),
partition o_ordk258 values less than (1133789062),
partition o_ordk259 values less than (1138183593),
partition o_ordk260 values less than (1142578124),
partition o_ordk261 values less than (1146972656),
partition o_ordk262 values less than (1151367187),
partition o_ordk263 values less than (1155761718),
partition o_ordk264 values less than (1160156249),
partition o_ordk265 values less than (1164550781),
partition o_ordk266 values less than (1168945312),
partition o_ordk267 values less than (1173339843),
partition o_ordk268 values less than (1177734374),
partition o_ordk269 values less than (1182128906),
partition o_ordk270 values less than (1186523437),
partition o_ordk271 values less than (1190917968),
partition o_ordk272 values less than (1195312499),
partition o_ordk273 values less than (1199707031),
partition o_ordk274 values less than (1204101562),
partition o_ordk275 values less than (1208496093),
partition o_ordk276 values less than (1212890624),
partition o_ordk277 values less than (1217285156),
partition o_ordk278 values less than (1221679687),
partition o_ordk279 values less than (1226074218),
partition o_ordk280 values less than (1230468749),
partition o_ordk281 values less than (1234863281),
partition o_ordk282 values less than (1239257812),
partition o_ordk283 values less than (1243652343),
partition o_ordk284 values less than (1248046874),
partition o_ordk285 values less than (1252441406),
partition o_ordk286 values less than (1256835937),
partition o_ordk287 values less than (1261230468),
partition o_ordk288 values less than (1265624999),
partition o_ordk289 values less than (1270019531),
partition o_ordk290 values less than (1274414062),
partition o_ordk291 values less than (1278808593),
partition o_ordk292 values less than (1283203124),
partition o_ordk293 values less than (1287597656),
partition o_ordk294 values less than (1291992187),
partition o_ordk295 values less than (1296386718),
partition o_ordk296 values less than (1300781249),
partition o_ordk297 values less than (1305175781),
partition o_ordk298 values less than (1309570312),
partition o_ordk299 values less than (1313964843),
partition o_ordk300 values less than (1318359374),
partition o_ordk301 values less than (1322753906),
partition o_ordk302 values less than (1327148437),
partition o_ordk303 values less than (1331542968),
partition o_ordk304 values less than (1335937499),
partition o_ordk305 values less than (1340332031),
partition o_ordk306 values less than (1344726562),
partition o_ordk307 values less than (1349121093),
partition o_ordk308 values less than (1353515624),
partition o_ordk309 values less than (1357910156),
partition o_ordk310 values less than (1362304687),
partition o_ordk311 values less than (1366699218),
partition o_ordk312 values less than (1371093749),
partition o_ordk313 values less than (1375488281),
partition o_ordk314 values less than (1379882812),
partition o_ordk315 values less than (1384277343),
partition o_ordk316 values less than (1388671874),
partition o_ordk317 values less than (1393066406),
partition o_ordk318 values less than (1397460937),
partition o_ordk319 values less than (1401855468),
partition o_ordk320 values less than (1406249999),
partition o_ordk321 values less than (1410644531),
partition o_ordk322 values less than (1415039062),
partition o_ordk323 values less than (1419433593),
partition o_ordk324 values less than (1423828124),
partition o_ordk325 values less than (1428222656),
partition o_ordk326 values less than (1432617187),
partition o_ordk327 values less than (1437011718),
partition o_ordk328 values less than (1441406249),
partition o_ordk329 values less than (1445800781),
partition o_ordk330 values less than (1450195312),
partition o_ordk331 values less than (1454589843),
partition o_ordk332 values less than (1458984374),
partition o_ordk333 values less than (1463378906),
partition o_ordk334 values less than (1467773437),
partition o_ordk335 values less than (1472167968),
partition o_ordk336 values less than (1476562499),
partition o_ordk337 values less than (1480957031),
partition o_ordk338 values less than (1485351562),
partition o_ordk339 values less than (1489746093),
partition o_ordk340 values less than (1494140624),
partition o_ordk341 values less than (1498535156),
partition o_ordk342 values less than (1502929687),
partition o_ordk343 values less than (1507324218),
partition o_ordk344 values less than (1511718749),
partition o_ordk345 values less than (1516113280),
partition o_ordk346 values less than (1520507812),
partition o_ordk347 values less than (1524902343),
partition o_ordk348 values less than (1529296874),
partition o_ordk349 values less than (1533691405),
partition o_ordk350 values less than (1538085937),
partition o_ordk351 values less than (1542480468),
partition o_ordk352 values less than (1546874999),
partition o_ordk353 values less than (1551269530),
partition o_ordk354 values less than (1555664062),
partition o_ordk355 values less than (1560058593),
partition o_ordk356 values less than (1564453124),
partition o_ordk357 values less than (1568847655),
partition o_ordk358 values less than (1573242187),
partition o_ordk359 values less than (1577636718),
partition o_ordk360 values less than (1582031249),
partition o_ordk361 values less than (1586425780),
partition o_ordk362 values less than (1590820312),
partition o_ordk363 values less than (1595214843),
partition o_ordk364 values less than (1599609374),
partition o_ordk365 values less than (1604003905),
partition o_ordk366 values less than (1608398437),
partition o_ordk367 values less than (1612792968),
partition o_ordk368 values less than (1617187499),
partition o_ordk369 values less than (1621582030),
partition o_ordk370 values less than (1625976562),
partition o_ordk371 values less than (1630371093),
partition o_ordk372 values less than (1634765624),
partition o_ordk373 values less than (1639160155),
partition o_ordk374 values less than (1643554687),
partition o_ordk375 values less than (1647949218),
partition o_ordk376 values less than (1652343749),
partition o_ordk377 values less than (1656738280),
partition o_ordk378 values less than (1661132812),
partition o_ordk379 values less than (1665527343),
partition o_ordk380 values less than (1669921874),
partition o_ordk381 values less than (1674316405),
partition o_ordk382 values less than (1678710937),
partition o_ordk383 values less than (1683105468),
partition o_ordk384 values less than (1687499999),
partition o_ordk385 values less than (1691894530),
partition o_ordk386 values less than (1696289062),
partition o_ordk387 values less than (1700683593),
partition o_ordk388 values less than (1705078124),
partition o_ordk389 values less than (1709472655),
partition o_ordk390 values less than (1713867187),
partition o_ordk391 values less than (1718261718),
partition o_ordk392 values less than (1722656249),
partition o_ordk393 values less than (1727050780),
partition o_ordk394 values less than (1731445312),
partition o_ordk395 values less than (1735839843),
partition o_ordk396 values less than (1740234374),
partition o_ordk397 values less than (1744628905),
partition o_ordk398 values less than (1749023437),
partition o_ordk399 values less than (1753417968),
partition o_ordk400 values less than (1757812499),
partition o_ordk401 values less than (1762207030),
partition o_ordk402 values less than (1766601562),
partition o_ordk403 values less than (1770996093),
partition o_ordk404 values less than (1775390624),
partition o_ordk405 values less than (1779785155),
partition o_ordk406 values less than (1784179687),
partition o_ordk407 values less than (1788574218),
partition o_ordk408 values less than (1792968749),
partition o_ordk409 values less than (1797363280),
partition o_ordk410 values less than (1801757812),
partition o_ordk411 values less than (1806152343),
partition o_ordk412 values less than (1810546874),
partition o_ordk413 values less than (1814941405),
partition o_ordk414 values less than (1819335937),
partition o_ordk415 values less than (1823730468),
partition o_ordk416 values less than (1828124999),
partition o_ordk417 values less than (1832519530),
partition o_ordk418 values less than (1836914062),
partition o_ordk419 values less than (1841308593),
partition o_ordk420 values less than (1845703124),
partition o_ordk421 values less than (1850097655),
partition o_ordk422 values less than (1854492187),
```

```
partition o_ordk423 values less than (1858886718),
partition o_ordk424 values less than (1863281249),
partition o_ordk425 values less than (1867675780),
partition o_ordk426 values less than (1872070312),
partition o_ordk427 values less than (1876464843),
partition o_ordk428 values less than (1880859374),
partition o_ordk429 values less than (1885253905),
partition o_ordk430 values less than (1889648437),
partition o_ordk431 values less than (1894042968),
partition o_ordk432 values less than (1898437499),
partition o_ordk433 values less than (1902832030),
partition o_ordk434 values less than (1907226562),
partition o_ordk435 values less than (1911621093),
partition o_ordk436 values less than (1916015624),
partition o_ordk437 values less than (1920410155),
partition o_ordk438 values less than (1924804687),
partition o_ordk439 values less than (1929199218),
partition o_ordk440 values less than (1933593749),
partition o_ordk441 values less than (1937988280),
partition o_ordk442 values less than (1942382812),
partition o_ordk443 values less than (1946777343),
partition o_ordk444 values less than (1951171874),
partition o_ordk445 values less than (1955566405),
partition o_ordk446 values less than (1959960937),
partition o_ordk447 values less than (1964355468),
partition o_ordk448 values less than (1968749999),
partition o_ordk449 values less than (1973144530),
partition o_ordk450 values less than (1977539062),
partition o_ordk451 values less than (1981933593),
partition o_ordk452 values less than (1986328124),
partition o_ordk453 values less than (1990722655),
partition o_ordk454 values less than (1995117187),
partition o_ordk455 values less than (1999511718),
partition o_ordk456 values less than (2003906249),
partition o_ordk457 values less than (2008300780),
partition o_ordk458 values less than (2012695311),
partition o_ordk459 values less than (2017089843),
partition o_ordk460 values less than (2021484374),
partition o_ordk461 values less than (2025878905),
partition o_ordk462 values less than (2030273436),
partition o_ordk463 values less than (2034667968),
partition o_ordk464 values less than (2039062499),
partition o_ordk465 values less than (2043457030),
partition o_ordk466 values less than (2047851561),
partition o_ordk467 values less than (2052246093),
partition o_ordk468 values less than (2056640624),
partition o_ordk469 values less than (2061035155),
partition o_ordk470 values less than (2065429686),
partition o_ordk471 values less than (2069824218),
partition o_ordk472 values less than (2074218749),
partition o_ordk473 values less than (2078613280),
partition o_ordk474 values less than (2083007811),
partition o_ordk475 values less than (2087402343),
partition o_ordk476 values less than (2091796874),
partition o_ordk477 values less than (2096191405),
partition o_ordk478 values less than (2100585936),
partition o_ordk479 values less than (2104980468),
partition o_ordk480 values less than (2109374999),
partition o_ordk481 values less than (2113769530),
partition o_ordk482 values less than (2118164061),
partition o_ordk483 values less than (2122558593),
partition o_ordk484 values less than (2126953124),
partition o_ordk485 values less than (2131347655),
partition o_ordk486 values less than (2135742186),
partition o_ordk487 values less than (2140136718),
partition o_ordk488 values less than (2144531249),
partition o_ordk489 values less than (2148925780),
partition o_ordk490 values less than (2153320311),
partition o_ordk491 values less than (2157714843),
partition o_ordk492 values less than (2162109374),
partition o_ordk493 values less than (2166503905),
partition o_ordk494 values less than (2170898436),
partition o_ordk495 values less than (2175292968),
partition o_ordk496 values less than (2179687499),
partition o_ordk497 values less than (2184082030),
partition o_ordk498 values less than (2188476561),
partition o_ordk499 values less than (2192871093),
partition o_ordk500 values less than (2197265624),
partition o_ordk501 values less than (2201660155),
partition o_ordk502 values less than (2206054686),
partition o_ordk503 values less than (2210449218),
partition o_ordk504 values less than (2214843749),
partition o_ordk505 values less than (2219238280),
partition o_ordk506 values less than (2223632811),
partition o_ordk507 values less than (2228027343),
partition o_ordk508 values less than (2232421874),
partition o_ordk509 values less than (2236816405),
partition o_ordk510 values less than (2241210936),
partition o_ordk511 values less than (2245605468),
partition o_ordk512 values less than (2249999999),
partition o_ordk513 values less than (2254394530),
partition o_ordk514 values less than (2258789061),
partition o_ordk515 values less than (2263183593),
partition o_ordk516 values less than (2267578124),
partition o_ordk517 values less than (2271972655),
partition o_ordk518 values less than (2276367186),
partition o_ordk519 values less than (2280761718),
partition o_ordk520 values less than (2285156249),
partition o_ordk521 values less than (2289550780),
partition o_ordk522 values less than (2293945311),
partition o_ordk523 values less than (2298339843),
partition o_ordk524 values less than (2302734374),
partition o_ordk525 values less than (2307128905),
partition o_ordk526 values less than (2311523436),
partition o_ordk527 values less than (2315917968),
partition o_ordk528 values less than (2320312499),
partition o_ordk529 values less than (2324707030),
partition o_ordk530 values less than (2329101561),
partition o_ordk531 values less than (2333496093),
partition o_ordk532 values less than (2337890624),
partition o_ordk533 values less than (2342285155),
partition o_ordk534 values less than (2346679686),
partition o_ordk535 values less than (2351074218),
partition o_ordk536 values less than (2355468749),
partition o_ordk537 values less than (2359863280),
partition o_ordk538 values less than (2364257811),
partition o_ordk539 values less than (2368652343),
partition o_ordk540 values less than (2373046874),
partition o_ordk541 values less than (2377441405),
partition o_ordk542 values less than (2381835936),
partition o_ordk543 values less than (2386230468),
partition o_ordk544 values less than (2390624999),
partition o_ordk545 values less than (2395019530),
partition o_ordk546 values less than (2399414061),
partition o_ordk547 values less than (2403808593),
partition o_ordk548 values less than (2408203124),
partition o_ordk549 values less than (2412597655),
partition o_ordk550 values less than (2416992186),
partition o_ordk551 values less than (2421386718),
partition o_ordk552 values less than (2425781249),
partition o_ordk553 values less than (2430175780),
partition o_ordk554 values less than (2434570311),
partition o_ordk555 values less than (2438964843),
partition o_ordk556 values less than (2443359374),
partition o_ordk557 values less than (2447753905),
partition o_ordk558 values less than (2452148436),
partition o_ordk559 values less than (2456542968),
partition o_ordk560 values less than (2460937499),
partition o_ordk561 values less than (2465332030),
partition o_ordk562 values less than (2469726561),
partition o_ordk563 values less than (2474121093),
partition o_ordk564 values less than (2478515624),
partition o_ordk565 values less than (2482910155),
partition o_ordk566 values less than (2487304686),
partition o_ordk567 values less than (2491699218),
partition o_ordk568 values less than (2496093749),
partition o_ordk569 values less than (2500488280),
partition o_ordk570 values less than (2504882811),
partition o_ordk571 values less than (2509277342),
partition o_ordk572 values less than (2513671874),
partition o_ordk573 values less than (2518066405),
partition o_ordk574 values less than (2522460936),
partition o_ordk575 values less than (2526855467),
partition o_ordk576 values less than (2531249999),
partition o_ordk577 values less than (2535644530),
partition o_ordk578 values less than (2540039061),
partition o_ordk579 values less than (2544433592),
partition o_ordk580 values less than (2548828124),
partition o_ordk581 values less than (2553222655),
partition o_ordk582 values less than (2557617186),
partition o_ordk583 values less than (2562011717),
partition o_ordk584 values less than (2566406249),
partition o_ordk585 values less than (2570800780),
partition o_ordk586 values less than (2575195311),
partition o_ordk587 values less than (2579589842),
partition o_ordk588 values less than (2583984374),
partition o_ordk589 values less than (2588378905),
partition o_ordk590 values less than (2592773436),
partition o_ordk591 values less than (2597167967),
partition o_ordk592 values less than (2601562499),
partition o_ordk593 values less than (2605957030),
partition o_ordk594 values less than (2610351561),
partition o_ordk595 values less than (2614746092),
partition o_ordk596 values less than (2619140624),
partition o_ordk597 values less than (2623535155),
partition o_ordk598 values less than (2627929686),
partition o_ordk599 values less than (2632324217),
partition o_ordk600 values less than (2636718749),
partition o_ordk601 values less than (2641113280),
partition o_ordk602 values less than (2645507811),
```

```
partition o_ordk603 values less than (2649902342),
partition o_ordk604 values less than (2654296874),
partition o_ordk605 values less than (2658691405),
partition o_ordk606 values less than (2663085936),
partition o_ordk607 values less than (2667480467),
partition o_ordk608 values less than (2671874999),
partition o_ordk609 values less than (2676269530),
partition o_ordk610 values less than (2680664061),
partition o_ordk611 values less than (2685058592),
partition o_ordk612 values less than (2689453124),
partition o_ordk613 values less than (2693847655),
partition o_ordk614 values less than (2698242186),
partition o_ordk615 values less than (2702636717),
partition o_ordk616 values less than (2707031249),
partition o_ordk617 values less than (2711425780),
partition o_ordk618 values less than (2715820311),
partition o_ordk619 values less than (2720214842),
partition o_ordk620 values less than (2724609374),
partition o_ordk621 values less than (2729003905),
partition o_ordk622 values less than (2733398436),
partition o_ordk623 values less than (2737792967),
partition o_ordk624 values less than (2742187499),
partition o_ordk625 values less than (2746582030),
partition o_ordk626 values less than (2750976561),
partition o_ordk627 values less than (2755371092),
partition o_ordk628 values less than (2759765624),
partition o_ordk629 values less than (2764160155),
partition o_ordk630 values less than (2768554686),
partition o_ordk631 values less than (2772949217),
partition o_ordk632 values less than (2777343749),
partition o_ordk633 values less than (2781738280),
partition o_ordk634 values less than (2786132811),
partition o_ordk635 values less than (2790527342),
partition o_ordk636 values less than (2794921874),
partition o_ordk637 values less than (2799316405),
partition o_ordk638 values less than (2803710936),
partition o_ordk639 values less than (2808105467),
partition o_ordk640 values less than (2812499999),
partition o_ordk641 values less than (2816894530),
partition o_ordk642 values less than (2821289061),
partition o_ordk643 values less than (2825683592),
partition o_ordk644 values less than (2830078124),
partition o_ordk645 values less than (2834472655),
partition o_ordk646 values less than (2838867186),
partition o_ordk647 values less than (2843261717),
partition o_ordk648 values less than (2847656249),
partition o_ordk649 values less than (2852050780),
partition o_ordk650 values less than (2856445311),
partition o_ordk651 values less than (2860839842),
partition o_ordk652 values less than (2865234374),
partition o_ordk653 values less than (2869628905),
partition o_ordk654 values less than (2874023436),
partition o_ordk655 values less than (2878417967),
partition o_ordk656 values less than (2882812499),
partition o_ordk657 values less than (2887207030),
partition o_ordk658 values less than (2891601561),
partition o_ordk659 values less than (2895996092),
partition o_ordk660 values less than (2900390624),
partition o_ordk661 values less than (2904785155),
partition o_ordk662 values less than (2909179686),
partition o_ordk663 values less than (2913574217),
partition o_ordk664 values less than (2917968749),
partition o_ordk665 values less than (2922363280),
partition o_ordk666 values less than (2926757811),
partition o_ordk667 values less than (2931152342),
partition o_ordk668 values less than (2935546874),
partition o_ordk669 values less than (2939941405),
partition o_ordk670 values less than (2944335936),
partition o_ordk671 values less than (2948730467),
partition o_ordk672 values less than (2953124999),
partition o_ordk673 values less than (2957519530),
partition o_ordk674 values less than (2961914061),
partition o_ordk675 values less than (2966308592),
partition o_ordk676 values less than (2970703124),
partition o_ordk677 values less than (2975097655),
partition o_ordk678 values less than (2979492186),
partition o_ordk679 values less than (2983886717),
partition o_ordk680 values less than (2988281249),
partition o_ordk681 values less than (2992675780),
partition o_ordk682 values less than (2997070311),
partition o_ordk683 values less than (3001464842),
partition o_ordk684 values less than (3005859373),
partition o_ordk685 values less than (3010253905),
partition o_ordk686 values less than (3014648436),
partition o_ordk687 values less than (3019042967),
partition o_ordk688 values less than (3023437498),
partition o_ordk689 values less than (3027832030),
partition o_ordk690 values less than (3032226561),
partition o_ordk691 values less than (3036621092),
partition o_ordk692 values less than (3041015623),
partition o_ordk693 values less than (3045410155),
partition o_ordk694 values less than (3049804686),
partition o_ordk695 values less than (3054199217),
partition o_ordk696 values less than (3058593748),
partition o_ordk697 values less than (3062988280),
partition o_ordk698 values less than (3067382811),
partition o_ordk699 values less than (3071777342),
partition o_ordk700 values less than (3076171873),
partition o_ordk701 values less than (3080566405),
partition o_ordk702 values less than (3084960936),
partition o_ordk703 values less than (3089355467),
partition o_ordk704 values less than (3093749998),
partition o_ordk705 values less than (3098144530),
partition o_ordk706 values less than (3102539061),
partition o_ordk707 values less than (3106933592),
partition o_ordk708 values less than (3111328123),
partition o_ordk709 values less than (3115722655),
partition o_ordk710 values less than (3120117186),
partition o_ordk711 values less than (3124511717),
partition o_ordk712 values less than (3128906248),
partition o_ordk713 values less than (3133300780),
partition o_ordk714 values less than (3137695311),
partition o_ordk715 values less than (3142089842),
partition o_ordk716 values less than (3146484373),
partition o_ordk717 values less than (3150878905),
partition o_ordk718 values less than (3155273436),
partition o_ordk719 values less than (3159667967),
partition o_ordk720 values less than (3164062498),
partition o_ordk721 values less than (3168457030),
partition o_ordk722 values less than (3172851561),
partition o_ordk723 values less than (3177246092),
partition o_ordk724 values less than (3181640623),
partition o_ordk725 values less than (3186035155),
partition o_ordk726 values less than (3190429686),
partition o_ordk727 values less than (3194824217),
partition o_ordk728 values less than (3199218748),
partition o_ordk729 values less than (3203613280),
partition o_ordk730 values less than (3208007811),
partition o_ordk731 values less than (3212402342),
partition o_ordk732 values less than (3216796873),
partition o_ordk733 values less than (3221191405),
partition o_ordk734 values less than (3225585936),
partition o_ordk735 values less than (3229980467),
partition o_ordk736 values less than (3234374998),
partition o_ordk737 values less than (3238769530),
partition o_ordk738 values less than (3243164061),
partition o_ordk739 values less than (3247558592),
partition o_ordk740 values less than (3251953123),
partition o_ordk741 values less than (3256347655),
partition o_ordk742 values less than (3260742186),
partition o_ordk743 values less than (3265136717),
partition o_ordk744 values less than (3269531248),
partition o_ordk745 values less than (3273925780),
partition o_ordk746 values less than (3278320311),
partition o_ordk747 values less than (3282714842),
partition o_ordk748 values less than (3287109373),
partition o_ordk749 values less than (3291503905),
partition o_ordk750 values less than (3295898436),
partition o_ordk751 values less than (3300292967),
partition o_ordk752 values less than (3304687498),
partition o_ordk753 values less than (3309082030),
partition o_ordk754 values less than (3313476561),
partition o_ordk755 values less than (3317871092),
partition o_ordk756 values less than (3322265623),
partition o_ordk757 values less than (3326660155),
partition o_ordk758 values less than (3331054686),
partition o_ordk759 values less than (3335449217),
partition o_ordk760 values less than (3339843748),
partition o_ordk761 values less than (3344238280),
partition o_ordk762 values less than (3348632811),
partition o_ordk763 values less than (3353027342),
partition o_ordk764 values less than (3357421873),
partition o_ordk765 values less than (3361816405),
partition o_ordk766 values less than (3366210936),
partition o_ordk767 values less than (3370605467),
partition o_ordk768 values less than (3374999998),
partition o_ordk769 values less than (3379394530),
partition o_ordk770 values less than (3383789061),
partition o_ordk771 values less than (3388183592),
partition o_ordk772 values less than (3392578123),
partition o_ordk773 values less than (3396972655),
partition o_ordk774 values less than (3401367186),
partition o_ordk775 values less than (3405761717),
partition o_ordk776 values less than (3410156248),
partition o_ordk777 values less than (3414550780),
partition o_ordk778 values less than (3418945311),
partition o_ordk779 values less than (3423339842),
partition o_ordk780 values less than (3427734373),
partition o_ordk781 values less than (3432128905),
partition o_ordk782 values less than (3436523436),
```

```
partition o_ordk783 values less than (3440917967),
partition o_ordk784 values less than (3445312498),
partition o_ordk785 values less than (3449707030),
partition o_ordk786 values less than (3454101561),
partition o_ordk787 values less than (3458496092),
partition o_ordk788 values less than (3462890623),
partition o_ordk789 values less than (3467285155),
partition o_ordk790 values less than (3471679686),
partition o_ordk791 values less than (3476074217),
partition o_ordk792 values less than (3480468748),
partition o_ordk793 values less than (3484863280),
partition o_ordk794 values less than (3489257811),
partition o_ordk795 values less than (3493652342),
partition o_ordk796 values less than (3498046873),
partition o_ordk797 values less than (3502441404),
partition o_ordk798 values less than (3506835936),
partition o_ordk799 values less than (3511230467),
partition o_ordk800 values less than (3515624998),
partition o_ordk801 values less than (3520019529),
partition o_ordk802 values less than (3524414061),
partition o_ordk803 values less than (3528808592),
partition o_ordk804 values less than (3533203123),
partition o_ordk805 values less than (3537597654),
partition o_ordk806 values less than (3541992186),
partition o_ordk807 values less than (3546386717),
partition o_ordk808 values less than (3550781248),
partition o_ordk809 values less than (3555175779),
partition o_ordk810 values less than (3559570311),
partition o_ordk811 values less than (3563964842),
partition o_ordk812 values less than (3568359373),
partition o_ordk813 values less than (3572753904),
partition o_ordk814 values less than (3577148436),
partition o_ordk815 values less than (3581542967),
partition o_ordk816 values less than (3585937498),
partition o_ordk817 values less than (3590332029),
partition o_ordk818 values less than (3594726561),
partition o_ordk819 values less than (3599121092),
partition o_ordk820 values less than (3603515623),
partition o_ordk821 values less than (3607910154),
partition o_ordk822 values less than (3612304686),
partition o_ordk823 values less than (3616699217),
partition o_ordk824 values less than (3621093748),
partition o_ordk825 values less than (3625488279),
partition o_ordk826 values less than (3629882811),
partition o_ordk827 values less than (3634277342),
partition o_ordk828 values less than (3638671873),
partition o_ordk829 values less than (3643066404),
partition o_ordk830 values less than (3647460936),
partition o_ordk831 values less than (3651855467),
partition o_ordk832 values less than (3656249998),
partition o_ordk833 values less than (3660644529),
partition o_ordk834 values less than (3665039061),
partition o_ordk835 values less than (3669433592),
partition o_ordk836 values less than (3673828123),
partition o_ordk837 values less than (3678222654),
partition o_ordk838 values less than (3682617186),
partition o_ordk839 values less than (3687011717),
partition o_ordk840 values less than (3691406248),
partition o_ordk841 values less than (3695800779),
partition o_ordk842 values less than (3700195311),
partition o_ordk843 values less than (3704589842),
partition o_ordk844 values less than (3708984373),
partition o_ordk845 values less than (3713378904),
partition o_ordk846 values less than (3717773436),
partition o_ordk847 values less than (3722167967),
partition o_ordk848 values less than (3726562498),
partition o_ordk849 values less than (3730957029),
partition o_ordk850 values less than (3735351561),
partition o_ordk851 values less than (3739746092),
partition o_ordk852 values less than (3744140623),
partition o_ordk853 values less than (3748535154),
partition o_ordk854 values less than (3752929686),
partition o_ordk855 values less than (3757324217),
partition o_ordk856 values less than (3761718748),
partition o_ordk857 values less than (3766113279),
partition o_ordk858 values less than (3770507811),
partition o_ordk859 values less than (3774902342),
partition o_ordk860 values less than (3779296873),
partition o_ordk861 values less than (3783691404),
partition o_ordk862 values less than (3788085936),
partition o_ordk863 values less than (3792480467),
partition o_ordk864 values less than (3796874998),
partition o_ordk865 values less than (3801269529),
partition o_ordk866 values less than (3805664061),
partition o_ordk867 values less than (3810058592),
partition o_ordk868 values less than (3814453123),
partition o_ordk869 values less than (3818847654),
partition o_ordk870 values less than (3823242186),
partition o_ordk871 values less than (3827636717),
partition o_ordk872 values less than (3832031248),
partition o_ordk873 values less than (3836425779),
partition o_ordk874 values less than (3840820311),
partition o_ordk875 values less than (3845214842),
partition o_ordk876 values less than (3849609373),
partition o_ordk877 values less than (3854003904),
partition o_ordk878 values less than (3858398436),
partition o_ordk879 values less than (3862792967),
partition o_ordk880 values less than (3867187498),
partition o_ordk881 values less than (3871582029),
partition o_ordk882 values less than (3875976561),
partition o_ordk883 values less than (3880371092),
partition o_ordk884 values less than (3884765623),
partition o_ordk885 values less than (3889160154),
partition o_ordk886 values less than (3893554686),
partition o_ordk887 values less than (3897949217),
partition o_ordk888 values less than (3902343748),
partition o_ordk889 values less than (3906738279),
partition o_ordk890 values less than (3911132811),
partition o_ordk891 values less than (3915527342),
partition o_ordk892 values less than (3919921873),
partition o_ordk893 values less than (3924316404),
partition o_ordk894 values less than (3928710936),
partition o_ordk895 values less than (3933105467),
partition o_ordk896 values less than (3937499998),
partition o_ordk897 values less than (3941894529),
partition o_ordk898 values less than (3946289061),
partition o_ordk899 values less than (3950683592),
partition o_ordk900 values less than (3955078123),
partition o_ordk901 values less than (3959472654),
partition o_ordk902 values less than (3963867186),
partition o_ordk903 values less than (3968261717),
partition o_ordk904 values less than (3972656248),
partition o_ordk905 values less than (3977050779),
partition o_ordk906 values less than (3981445311),
partition o_ordk907 values less than (3985839842),
partition o_ordk908 values less than (3990234373),
partition o_ordk909 values less than (3994628904),
partition o_ordk910 values less than (3999023436),
partition o_ordk911 values less than (4003417967),
partition o_ordk912 values less than (4007812498),
partition o_ordk913 values less than (4012207029),
partition o_ordk914 values less than (4016601560),
partition o_ordk915 values less than (4020996092),
partition o_ordk916 values less than (4025390623),
partition o_ordk917 values less than (4029785154),
partition o_ordk918 values less than (4034179685),
partition o_ordk919 values less than (4038574217),
partition o_ordk920 values less than (4042968748),
partition o_ordk921 values less than (4047363279),
partition o_ordk922 values less than (4051757810),
partition o_ordk923 values less than (4056152342),
partition o_ordk924 values less than (4060546873),
partition o_ordk925 values less than (4064941404),
partition o_ordk926 values less than (4069335935),
partition o_ordk927 values less than (4073730467),
partition o_ordk928 values less than (4078124998),
partition o_ordk929 values less than (4082519529),
partition o_ordk930 values less than (4086914060),
partition o_ordk931 values less than (4091308592),
partition o_ordk932 values less than (4095703123),
partition o_ordk933 values less than (4100097654),
partition o_ordk934 values less than (4104492185),
partition o_ordk935 values less than (4108886717),
partition o_ordk936 values less than (4113281248),
partition o_ordk937 values less than (4117675779),
partition o_ordk938 values less than (4122070310),
partition o_ordk939 values less than (4126464842),
partition o_ordk940 values less than (4130859373),
partition o_ordk941 values less than (4135253904),
partition o_ordk942 values less than (4139648435),
partition o_ordk943 values less than (4144042967),
partition o_ordk944 values less than (4148437498),
partition o_ordk945 values less than (4152832029),
partition o_ordk946 values less than (4157226560),
partition o_ordk947 values less than (4161621092),
partition o_ordk948 values less than (4166015623),
partition o_ordk949 values less than (4170410154),
partition o_ordk950 values less than (4174804685),
partition o_ordk951 values less than (4179199217),
partition o_ordk952 values less than (4183593748),
partition o_ordk953 values less than (4187988279),
partition o_ordk954 values less than (4192382810),
partition o_ordk955 values less than (4196777342),
partition o_ordk956 values less than (4201171873),
partition o_ordk957 values less than (4205566404),
partition o_ordk958 values less than (4209960935),
partition o_ordk959 values less than (4214355467),
partition o_ordk960 values less than (4218749998),
partition o_ordk961 values less than (4223144529),
partition o_ordk962 values less than (4227539060),
```

```
partition o_ordk963 values less than (4231933592),
partition o_ordk964 values less than (4236328123),
partition o_ordk965 values less than (4240722654),
partition o_ordk966 values less than (4245117185),
partition o_ordk967 values less than (4249511717),
partition o_ordk968 values less than (4253906248),
partition o_ordk969 values less than (4258300779),
partition o_ordk970 values less than (4262695310),
partition o_ordk971 values less than (4267089842),
partition o_ordk972 values less than (4271484373),
partition o_ordk973 values less than (4275878904),
partition o_ordk974 values less than (4280273435),
partition o_ordk975 values less than (4284667967),
partition o_ordk976 values less than (4289062498),
partition o_ordk977 values less than (4293457029),
partition o_ordk978 values less than (4297851560),
partition o_ordk979 values less than (4302246092),
partition o_ordk980 values less than (4306640623),
partition o_ordk981 values less than (4311035154),
partition o_ordk982 values less than (4315429685),
partition o_ordk983 values less than (4319824217),
partition o_ordk984 values less than (4324218748),
partition o_ordk985 values less than (4328613279),
partition o_ordk986 values less than (4333007810),
partition o_ordk987 values less than (4337402342),
partition o_ordk988 values less than (4341796873),
partition o_ordk989 values less than (4346191404),
partition o_ordk990 values less than (4350585935),
partition o_ordk991 values less than (4354980467),
partition o_ordk992 values less than (4359374998),
partition o_ordk993 values less than (4363769529),
partition o_ordk994 values less than (4368164060),
partition o_ordk995 values less than (4372558592),
partition o_ordk996 values less than (4376953123),
partition o_ordk997 values less than (4381347654),
partition o_ordk998 values less than (4385742185),
partition o_ordk999 values less than (4390136717),
partition o_ordk1000 values less than (4394531248),
partition o_ordk1001 values less than (4398925779),
partition o_ordk1002 values less than (4403320310),
partition o_ordk1003 values less than (4407714842),
partition o_ordk1004 values less than (4412109373),
partition o_ordk1005 values less than (4416503904),
partition o_ordk1006 values less than (4420898435),
partition o_ordk1007 values less than (4425292967),
partition o_ordk1008 values less than (4429687498),
partition o_ordk1009 values less than (4434082029),
partition o_ordk1010 values less than (4438476560),
partition o_ordk1011 values less than (4442871092),
partition o_ordk1012 values less than (4447265623),
partition o_ordk1013 values less than (4451660154),
partition o_ordk1014 values less than (4456054685),
partition o_ordk1015 values less than (4460449217),
partition o_ordk1016 values less than (4464843748),
partition o_ordk1017 values less than (4469238279),
partition o_ordk1018 values less than (4473632810),
partition o_ordk1019 values less than (4478027342),
partition o_ordk1020 values less than (4482421873),
partition o_ordk1021 values less than (4486816404),
partition o_ordk1022 values less than (4491210935),
partition o_ordk1023 values less than (4495605467),
partition o_ordk1024 values less than (4499999998),
partition o_ordk1025 values less than (4504394529),
partition o_ordk1026 values less than (4508789060),
partition o_ordk1027 values less than (4513183591),
partition o_ordk1028 values less than (4517578123),
partition o_ordk1029 values less than (4521972654),
partition o_ordk1030 values less than (4526367185),
partition o_ordk1031 values less than (4530761716),
partition o_ordk1032 values less than (4535156248),
partition o_ordk1033 values less than (4539550779),
partition o_ordk1034 values less than (4543945310),
partition o_ordk1035 values less than (4548339841),
partition o_ordk1036 values less than (4552734373),
partition o_ordk1037 values less than (4557128904),
partition o_ordk1038 values less than (4561523435),
partition o_ordk1039 values less than (4565917966),
partition o_ordk1040 values less than (4570312498),
partition o_ordk1041 values less than (4574707029),
partition o_ordk1042 values less than (4579101560),
partition o_ordk1043 values less than (4583496091),
partition o_ordk1044 values less than (4587890623),
partition o_ordk1045 values less than (4592285154),
partition o_ordk1046 values less than (4596679685),
partition o_ordk1047 values less than (4601074216),
partition o_ordk1048 values less than (4605468748),
partition o_ordk1049 values less than (4609863279),
partition o_ordk1050 values less than (4614257810),
partition o_ordk1051 values less than (4618652341),
partition o_ordk1052 values less than (4623046873),
partition o_ordk1053 values less than (4627441404),
partition o_ordk1054 values less than (4631835935),
partition o_ordk1055 values less than (4636230466),
partition o_ordk1056 values less than (4640624998),
partition o_ordk1057 values less than (4645019529),
partition o_ordk1058 values less than (4649414060),
partition o_ordk1059 values less than (4653808591),
partition o_ordk1060 values less than (4658203123),
partition o_ordk1061 values less than (4662597654),
partition o_ordk1062 values less than (4666992185),
partition o_ordk1063 values less than (4671386716),
partition o_ordk1064 values less than (4675781248),
partition o_ordk1065 values less than (4680175779),
partition o_ordk1066 values less than (4684570310),
partition o_ordk1067 values less than (4688964841),
partition o_ordk1068 values less than (4693359373),
partition o_ordk1069 values less than (4697753904),
partition o_ordk1070 values less than (4702148435),
partition o_ordk1071 values less than (4706542966),
partition o_ordk1072 values less than (4710937498),
partition o_ordk1073 values less than (4715332029),
partition o_ordk1074 values less than (4719726560),
partition o_ordk1075 values less than (4724121091),
partition o_ordk1076 values less than (4728515623),
partition o_ordk1077 values less than (4732910154),
partition o_ordk1078 values less than (4737304685),
partition o_ordk1079 values less than (4741699216),
partition o_ordk1080 values less than (4746093748),
partition o_ordk1081 values less than (4750488279),
partition o_ordk1082 values less than (4754882810),
partition o_ordk1083 values less than (4759277341),
partition o_ordk1084 values less than (4763671873),
partition o_ordk1085 values less than (4768066404),
partition o_ordk1086 values less than (4772460935),
partition o_ordk1087 values less than (4776855466),
partition o_ordk1088 values less than (4781249998),
partition o_ordk1089 values less than (4785644529),
partition o_ordk1090 values less than (4790039060),
partition o_ordk1091 values less than (4794433591),
partition o_ordk1092 values less than (4798828123),
partition o_ordk1093 values less than (4803222654),
partition o_ordk1094 values less than (4807617185),
partition o_ordk1095 values less than (4812011716),
partition o_ordk1096 values less than (4816406248),
partition o_ordk1097 values less than (4820800779),
partition o_ordk1098 values less than (4825195310),
partition o_ordk1099 values less than (4829589841),
partition o_ordk1100 values less than (4833984373),
partition o_ordk1101 values less than (4838378904),
partition o_ordk1102 values less than (4842773435),
partition o_ordk1103 values less than (4847167966),
partition o_ordk1104 values less than (4851562498),
partition o_ordk1105 values less than (4855957029),
partition o_ordk1106 values less than (4860351560),
partition o_ordk1107 values less than (4864746091),
partition o_ordk1108 values less than (4869140623),
partition o_ordk1109 values less than (4873535154),
partition o_ordk1110 values less than (4877929685),
partition o_ordk1111 values less than (4882324216),
partition o_ordk1112 values less than (4886718748),
partition o_ordk1113 values less than (4891113279),
partition o_ordk1114 values less than (4895507810),
partition o_ordk1115 values less than (4899902341),
partition o_ordk1116 values less than (4904296873),
partition o_ordk1117 values less than (4908691404),
partition o_ordk1118 values less than (4913085935),
partition o_ordk1119 values less than (4917480466),
partition o_ordk1120 values less than (4921874998),
partition o_ordk1121 values less than (4926269529),
partition o_ordk1122 values less than (4930664060),
partition o_ordk1123 values less than (4935058591),
partition o_ordk1124 values less than (4939453123),
partition o_ordk1125 values less than (4943847654),
partition o_ordk1126 values less than (4948242185),
partition o_ordk1127 values less than (4952636716),
partition o_ordk1128 values less than (4957031248),
partition o_ordk1129 values less than (4961425779),
partition o_ordk1130 values less than (4965820310),
partition o_ordk1131 values less than (4970214841),
partition o_ordk1132 values less than (4974609373),
partition o_ordk1133 values less than (4979003904),
partition o_ordk1134 values less than (4983398435),
partition o_ordk1135 values less than (4987792966),
partition o_ordk1136 values less than (4992187498),
partition o_ordk1137 values less than (4996582029),
partition o_ordk1138 values less than (5000976560),
partition o_ordk1139 values less than (5005371091),
partition o_ordk1140 values less than (5009765622),
partition o_ordk1141 values less than (5014160154),
partition o_ordk1142 values less than (5018554685),
```

```
partition o_ordk1143 values less than (5022949216),
partition o_ordk1144 values less than (5027343747),
partition o_ordk1145 values less than (5031738279),
partition o_ordk1146 values less than (5036132810),
partition o_ordk1147 values less than (5040527341),
partition o_ordk1148 values less than (5044921872),
partition o_ordk1149 values less than (5049316404),
partition o_ordk1150 values less than (5053710935),
partition o_ordk1151 values less than (5058105466),
partition o_ordk1152 values less than (5062499997),
partition o_ordk1153 values less than (5066894529),
partition o_ordk1154 values less than (5071289060),
partition o_ordk1155 values less than (5075683591),
partition o_ordk1156 values less than (5080078122),
partition o_ordk1157 values less than (5084472654),
partition o_ordk1158 values less than (5088867185),
partition o_ordk1159 values less than (5093261716),
partition o_ordk1160 values less than (5097656247),
partition o_ordk1161 values less than (5102050779),
partition o_ordk1162 values less than (5106445310),
partition o_ordk1163 values less than (5110839841),
partition o_ordk1164 values less than (5115234372),
partition o_ordk1165 values less than (5119628904),
partition o_ordk1166 values less than (5124023435),
partition o_ordk1167 values less than (5128417966),
partition o_ordk1168 values less than (5132812497),
partition o_ordk1169 values less than (5137207029),
partition o_ordk1170 values less than (5141601560),
partition o_ordk1171 values less than (5145996091),
partition o_ordk1172 values less than (5150390622),
partition o_ordk1173 values less than (5154785154),
partition o_ordk1174 values less than (5159179685),
partition o_ordk1175 values less than (5163574216),
partition o_ordk1176 values less than (5167968747),
partition o_ordk1177 values less than (5172363279),
partition o_ordk1178 values less than (5176757810),
partition o_ordk1179 values less than (5181152341),
partition o_ordk1180 values less than (5185546872),
partition o_ordk1181 values less than (5189941404),
partition o_ordk1182 values less than (5194335935),
partition o_ordk1183 values less than (5198730466),
partition o_ordk1184 values less than (5203124997),
partition o_ordk1185 values less than (5207519529),
partition o_ordk1186 values less than (5211914060),
partition o_ordk1187 values less than (5216308591),
partition o_ordk1188 values less than (5220703122),
partition o_ordk1189 values less than (5225097654),
partition o_ordk1190 values less than (5229492185),
partition o_ordk1191 values less than (5233886716),
partition o_ordk1192 values less than (5238281247),
partition o_ordk1193 values less than (5242675779),
partition o_ordk1194 values less than (5247070310),
partition o_ordk1195 values less than (5251464841),
partition o_ordk1196 values less than (5255859372),
partition o_ordk1197 values less than (5260253904),
partition o_ordk1198 values less than (5264648435),
partition o_ordk1199 values less than (5269042966),
partition o_ordk1200 values less than (5273437497),
partition o_ordk1201 values less than (5277832029),
partition o_ordk1202 values less than (5282226560),
partition o_ordk1203 values less than (5286621091),
partition o_ordk1204 values less than (5291015622),
partition o_ordk1205 values less than (5295410154),
partition o_ordk1206 values less than (5299804685),
partition o_ordk1207 values less than (5304199216),
partition o_ordk1208 values less than (5308593747),
partition o_ordk1209 values less than (5312988279),
partition o_ordk1210 values less than (5317382810),
partition o_ordk1211 values less than (5321777341),
partition o_ordk1212 values less than (5326171872),
partition o_ordk1213 values less than (5330566404),
partition o_ordk1214 values less than (5334960935),
partition o_ordk1215 values less than (5339355466),
partition o_ordk1216 values less than (5343749997),
partition o_ordk1217 values less than (5348144529),
partition o_ordk1218 values less than (5352539060),
partition o_ordk1219 values less than (5356933591),
partition o_ordk1220 values less than (5361328122),
partition o_ordk1221 values less than (5365722654),
partition o_ordk1222 values less than (5370117185),
partition o_ordk1223 values less than (5374511716),
partition o_ordk1224 values less than (5378906247),
partition o_ordk1225 values less than (5383300779),
partition o_ordk1226 values less than (5387695310),
partition o_ordk1227 values less than (5392089841),
partition o_ordk1228 values less than (5396484372),
partition o_ordk1229 values less than (5400878904),
partition o_ordk1230 values less than (5405273435),
partition o_ordk1231 values less than (5409667966),
partition o_ordk1232 values less than (5414062497),
partition o_ordk1233 values less than (5418457029),
partition o_ordk1234 values less than (5422851560),
partition o_ordk1235 values less than (5427246091),
partition o_ordk1236 values less than (5431640622),
partition o_ordk1237 values less than (5436035154),
partition o_ordk1238 values less than (5440429685),
partition o_ordk1239 values less than (5444824216),
partition o_ordk1240 values less than (5449218747),
partition o_ordk1241 values less than (5453613279),
partition o_ordk1242 values less than (5458007810),
partition o_ordk1243 values less than (5462402341),
partition o_ordk1244 values less than (5466796872),
partition o_ordk1245 values less than (5471191404),
partition o_ordk1246 values less than (5475585935),
partition o_ordk1247 values less than (5479980466),
partition o_ordk1248 values less than (5484374997),
partition o_ordk1249 values less than (5488769529),
partition o_ordk1250 values less than (5493164060),
partition o_ordk1251 values less than (5497558591),
partition o_ordk1252 values less than (5501953122),
partition o_ordk1253 values less than (5506347653),
partition o_ordk1254 values less than (5510742185),
partition o_ordk1255 values less than (5515136716),
partition o_ordk1256 values less than (5519531247),
partition o_ordk1257 values less than (5523925778),
partition o_ordk1258 values less than (5528320310),
partition o_ordk1259 values less than (5532714841),
partition o_ordk1260 values less than (5537109372),
partition o_ordk1261 values less than (5541503903),
partition o_ordk1262 values less than (5545898435),
partition o_ordk1263 values less than (5550292966),
partition o_ordk1264 values less than (5554687497),
partition o_ordk1265 values less than (5559082028),
partition o_ordk1266 values less than (5563476560),
partition o_ordk1267 values less than (5567871091),
partition o_ordk1268 values less than (5572265622),
partition o_ordk1269 values less than (5576660153),
partition o_ordk1270 values less than (5581054685),
partition o_ordk1271 values less than (5585449216),
partition o_ordk1272 values less than (5589843747),
partition o_ordk1273 values less than (5594238278),
partition o_ordk1274 values less than (5598632810),
partition o_ordk1275 values less than (5603027341),
partition o_ordk1276 values less than (5607421872),
partition o_ordk1277 values less than (5611816403),
partition o_ordk1278 values less than (5616210935),
partition o_ordk1279 values less than (5620605466),
partition o_ordk1280 values less than (5624999997),
partition o_ordk1281 values less than (5629394528),
partition o_ordk1282 values less than (5633789060),
partition o_ordk1283 values less than (5638183591),
partition o_ordk1284 values less than (5642578122),
partition o_ordk1285 values less than (5646972653),
partition o_ordk1286 values less than (5651367185),
partition o_ordk1287 values less than (5655761716),
partition o_ordk1288 values less than (5660156247),
partition o_ordk1289 values less than (5664550778),
partition o_ordk1290 values less than (5668945310),
partition o_ordk1291 values less than (5673339841),
partition o_ordk1292 values less than (5677734372),
partition o_ordk1293 values less than (5682128903),
partition o_ordk1294 values less than (5686523435),
partition o_ordk1295 values less than (5690917966),
partition o_ordk1296 values less than (5695312497),
partition o_ordk1297 values less than (5699707028),
partition o_ordk1298 values less than (5704101560),
partition o_ordk1299 values less than (5708496091),
partition o_ordk1300 values less than (5712890622),
partition o_ordk1301 values less than (5717285153),
partition o_ordk1302 values less than (5721679685),
partition o_ordk1303 values less than (5726074216),
partition o_ordk1304 values less than (5730468747),
partition o_ordk1305 values less than (5734863278),
partition o_ordk1306 values less than (5739257810),
partition o_ordk1307 values less than (5743652341),
partition o_ordk1308 values less than (5748046872),
partition o_ordk1309 values less than (5752441403),
partition o_ordk1310 values less than (5756835935),
partition o_ordk1311 values less than (5761230466),
partition o_ordk1312 values less than (5765624997),
partition o_ordk1313 values less than (5770019528),
partition o_ordk1314 values less than (5774414060),
partition o_ordk1315 values less than (5778808591),
partition o_ordk1316 values less than (5783203122),
partition o_ordk1317 values less than (5787597653),
partition o_ordk1318 values less than (5791992185),
partition o_ordk1319 values less than (5796386716),
partition o_ordk1320 values less than (5800781247),
partition o_ordk1321 values less than (5805175778),
partition o_ordk1322 values less than (5809570310),
```

```
partition o_ordk1323 values less than (5813964841),
partition o_ordk1324 values less than (5818359372),
partition o_ordk1325 values less than (5822753903),
partition o_ordk1326 values less than (5827148435),
partition o_ordk1327 values less than (5831542966),
partition o_ordk1328 values less than (5835937497),
partition o_ordk1329 values less than (5840332028),
partition o_ordk1330 values less than (5844726560),
partition o_ordk1331 values less than (5849121091),
partition o_ordk1332 values less than (5853515622),
partition o_ordk1333 values less than (5857910153),
partition o_ordk1334 values less than (5862304685),
partition o_ordk1335 values less than (5866699216),
partition o_ordk1336 values less than (5871093747),
partition o_ordk1337 values less than (5875488278),
partition o_ordk1338 values less than (5879882810),
partition o_ordk1339 values less than (5884277341),
partition o_ordk1340 values less than (5888671872),
partition o_ordk1341 values less than (5893066403),
partition o_ordk1342 values less than (5897460935),
partition o_ordk1343 values less than (5901855466),
partition o_ordk1344 values less than (5906249997),
partition o_ordk1345 values less than (5910644528),
partition o_ordk1346 values less than (5915039060),
partition o_ordk1347 values less than (5919433591),
partition o_ordk1348 values less than (5923828122),
partition o_ordk1349 values less than (5928222653),
partition o_ordk1350 values less than (5932617185),
partition o_ordk1351 values less than (5937011716),
partition o_ordk1352 values less than (5941406247),
partition o_ordk1353 values less than (5945800778),
partition o_ordk1354 values less than (5950195310),
partition o_ordk1355 values less than (5954589841),
partition o_ordk1356 values less than (5958984372),
partition o_ordk1357 values less than (5963378903),
partition o_ordk1358 values less than (5967773435),
partition o_ordk1359 values less than (5972167966),
partition o_ordk1360 values less than (5976562497),
partition o_ordk1361 values less than (5980957028),
partition o_ordk1362 values less than (5985351560),
partition o_ordk1363 values less than (5989746091),
partition o_ordk1364 values less than (5994140622),
partition o_ordk1365 values less than (5998535153),
partition o_ordk1366 values less than (6002929684),
partition o_ordk1367 values less than (6007324216),
partition o_ordk1368 values less than (6011718747),
partition o_ordk1369 values less than (6016113278),
partition o_ordk1370 values less than (6020507809),
partition o_ordk1371 values less than (6024902341),
partition o_ordk1372 values less than (6029296872),
partition o_ordk1373 values less than (6033691403),
partition o_ordk1374 values less than (6038085934),
partition o_ordk1375 values less than (6042480466),
partition o_ordk1376 values less than (6046874997),
partition o_ordk1377 values less than (6051269528),
partition o_ordk1378 values less than (6055664059),
partition o_ordk1379 values less than (6060058591),
partition o_ordk1380 values less than (6064453122),
partition o_ordk1381 values less than (6068847653),
partition o_ordk1382 values less than (6073242184),
partition o_ordk1383 values less than (6077636716),
partition o_ordk1384 values less than (6082031247),
partition o_ordk1385 values less than (6086425778),
partition o_ordk1386 values less than (6090820309),
partition o_ordk1387 values less than (6095214841),
partition o_ordk1388 values less than (6099609372),
partition o_ordk1389 values less than (6104003903),
partition o_ordk1390 values less than (6108398434),
partition o_ordk1391 values less than (6112792966),
partition o_ordk1392 values less than (6117187497),
partition o_ordk1393 values less than (6121582028),
partition o_ordk1394 values less than (6125976559),
partition o_ordk1395 values less than (6130371091),
partition o_ordk1396 values less than (6134765622),
partition o_ordk1397 values less than (6139160153),
partition o_ordk1398 values less than (6143554684),
partition o_ordk1399 values less than (6147949216),
partition o_ordk1400 values less than (6152343747),
partition o_ordk1401 values less than (6156738278),
partition o_ordk1402 values less than (6161132809),
partition o_ordk1403 values less than (6165527341),
partition o_ordk1404 values less than (6169921872),
partition o_ordk1405 values less than (6174316403),
partition o_ordk1406 values less than (6178710934),
partition o_ordk1407 values less than (6183105466),
partition o_ordk1408 values less than (6187499997),
partition o_ordk1409 values less than (6191894528),
partition o_ordk1410 values less than (6196289059),
partition o_ordk1411 values less than (6200683591),
partition o_ordk1412 values less than (6205078122),
partition o_ordk1413 values less than (6209472653),
partition o_ordk1414 values less than (6213867184),
partition o_ordk1415 values less than (6218261716),
partition o_ordk1416 values less than (6222656247),
partition o_ordk1417 values less than (6227050778),
partition o_ordk1418 values less than (6231445309),
partition o_ordk1419 values less than (6235839841),
partition o_ordk1420 values less than (6240234372),
partition o_ordk1421 values less than (6244628903),
partition o_ordk1422 values less than (6249023434),
partition o_ordk1423 values less than (6253417966),
partition o_ordk1424 values less than (6257812497),
partition o_ordk1425 values less than (6262207028),
partition o_ordk1426 values less than (6266601559),
partition o_ordk1427 values less than (6270996091),
partition o_ordk1428 values less than (6275390622),
partition o_ordk1429 values less than (6279785153),
partition o_ordk1430 values less than (6284179684),
partition o_ordk1431 values less than (6288574216),
partition o_ordk1432 values less than (6292968747),
partition o_ordk1433 values less than (6297363278),
partition o_ordk1434 values less than (6301757809),
partition o_ordk1435 values less than (6306152341),
partition o_ordk1436 values less than (6310546872),
partition o_ordk1437 values less than (6314941403),
partition o_ordk1438 values less than (6319335934),
partition o_ordk1439 values less than (6323730466),
partition o_ordk1440 values less than (6328124997),
partition o_ordk1441 values less than (6332519528),
partition o_ordk1442 values less than (6336914059),
partition o_ordk1443 values less than (6341308591),
partition o_ordk1444 values less than (6345703122),
partition o_ordk1445 values less than (6350097653),
partition o_ordk1446 values less than (6354492184),
partition o_ordk1447 values less than (6358886716),
partition o_ordk1448 values less than (6363281247),
partition o_ordk1449 values less than (6367675778),
partition o_ordk1450 values less than (6372070309),
partition o_ordk1451 values less than (6376464841),
partition o_ordk1452 values less than (6380859372),
partition o_ordk1453 values less than (6385253903),
partition o_ordk1454 values less than (6389648434),
partition o_ordk1455 values less than (6394042966),
partition o_ordk1456 values less than (6398437497),
partition o_ordk1457 values less than (6402832028),
partition o_ordk1458 values less than (6407226559),
partition o_ordk1459 values less than (6411621091),
partition o_ordk1460 values less than (6416015622),
partition o_ordk1461 values less than (6420410153),
partition o_ordk1462 values less than (6424804684),
partition o_ordk1463 values less than (6429199216),
partition o_ordk1464 values less than (6433593747),
partition o_ordk1465 values less than (6437988278),
partition o_ordk1466 values less than (6442382809),
partition o_ordk1467 values less than (6446777341),
partition o_ordk1468 values less than (6451171872),
partition o_ordk1469 values less than (6455566403),
partition o_ordk1470 values less than (6459960934),
partition o_ordk1471 values less than (6464355466),
partition o_ordk1472 values less than (6468749997),
partition o_ordk1473 values less than (6473144528),
partition o_ordk1474 values less than (6477539059),
partition o_ordk1475 values less than (6481933591),
partition o_ordk1476 values less than (6486328122),
partition o_ordk1477 values less than (6490722653),
partition o_ordk1478 values less than (6495117184),
partition o_ordk1479 values less than (6499511716),
partition o_ordk1480 values less than (6503906247),
partition o_ordk1481 values less than (6508300778),
partition o_ordk1482 values less than (6512695309),
partition o_ordk1483 values less than (6517089840),
partition o_ordk1484 values less than (6521484372),
partition o_ordk1485 values less than (6525878903),
partition o_ordk1486 values less than (6530273434),
partition o_ordk1487 values less than (6534667965),
partition o_ordk1488 values less than (6539062497),
partition o_ordk1489 values less than (6543457028),
partition o_ordk1490 values less than (6547851559),
partition o_ordk1491 values less than (6552246090),
partition o_ordk1492 values less than (6556640622),
partition o_ordk1493 values less than (6561035153),
partition o_ordk1494 values less than (6565429684),
partition o_ordk1495 values less than (6569824215),
partition o_ordk1496 values less than (6574218747),
partition o_ordk1497 values less than (6578613278),
partition o_ordk1498 values less than (6583007809),
partition o_ordk1499 values less than (6587402340),
partition o_ordk1500 values less than (6591796872),
partition o_ordk1501 values less than (6596191403),
partition o_ordk1502 values less than (6600585934),
```

```
partition o_ordk1503 values less than (6604980465),
partition o_ordk1504 values less than (6609374997),
partition o_ordk1505 values less than (6613769528),
partition o_ordk1506 values less than (6618164059),
partition o_ordk1507 values less than (6622558590),
partition o_ordk1508 values less than (6626953122),
partition o_ordk1509 values less than (6631347653),
partition o_ordk1510 values less than (6635742184),
partition o_ordk1511 values less than (6640136715),
partition o_ordk1512 values less than (6644531247),
partition o_ordk1513 values less than (6648925778),
partition o_ordk1514 values less than (6653320309),
partition o_ordk1515 values less than (6657714840),
partition o_ordk1516 values less than (6662109372),
partition o_ordk1517 values less than (6666503903),
partition o_ordk1518 values less than (6670898434),
partition o_ordk1519 values less than (6675292965),
partition o_ordk1520 values less than (6679687497),
partition o_ordk1521 values less than (6684082028),
partition o_ordk1522 values less than (6688476559),
partition o_ordk1523 values less than (6692871090),
partition o_ordk1524 values less than (6697265622),
partition o_ordk1525 values less than (6701660153),
partition o_ordk1526 values less than (6706054684),
partition o_ordk1527 values less than (6710449215),
partition o_ordk1528 values less than (6714843747),
partition o_ordk1529 values less than (6719238278),
partition o_ordk1530 values less than (6723632809),
partition o_ordk1531 values less than (6728027340),
partition o_ordk1532 values less than (6732421872),
partition o_ordk1533 values less than (6736816403),
partition o_ordk1534 values less than (6741210934),
partition o_ordk1535 values less than (6745605465),
partition o_ordk1536 values less than (6749999997),
partition o_ordk1537 values less than (6754394528),
partition o_ordk1538 values less than (6758789059),
partition o_ordk1539 values less than (6763183590),
partition o_ordk1540 values less than (6767578122),
partition o_ordk1541 values less than (6771972653),
partition o_ordk1542 values less than (6776367184),
partition o_ordk1543 values less than (6780761715),
partition o_ordk1544 values less than (6785156247),
partition o_ordk1545 values less than (6789550778),
partition o_ordk1546 values less than (6793945309),
partition o_ordk1547 values less than (6798339840),
partition o_ordk1548 values less than (6802734372),
partition o_ordk1549 values less than (6807128903),
partition o_ordk1550 values less than (6811523434),
partition o_ordk1551 values less than (6815917965),
partition o_ordk1552 values less than (6820312497),
partition o_ordk1553 values less than (6824707028),
partition o_ordk1554 values less than (6829101559),
partition o_ordk1555 values less than (6833496090),
partition o_ordk1556 values less than (6837890622),
partition o_ordk1557 values less than (6842285153),
partition o_ordk1558 values less than (6846679684),
partition o_ordk1559 values less than (6851074215),
partition o_ordk1560 values less than (6855468747),
partition o_ordk1561 values less than (6859863278),
partition o_ordk1562 values less than (6864257809),
partition o_ordk1563 values less than (6868652340),
partition o_ordk1564 values less than (6873046872),
partition o_ordk1565 values less than (6877441403),
partition o_ordk1566 values less than (6881835934),
partition o_ordk1567 values less than (6886230465),
partition o_ordk1568 values less than (6890624997),
partition o_ordk1569 values less than (6895019528),
partition o_ordk1570 values less than (6899414059),
partition o_ordk1571 values less than (6903808590),
partition o_ordk1572 values less than (6908203122),
partition o_ordk1573 values less than (6912597653),
partition o_ordk1574 values less than (6916992184),
partition o_ordk1575 values less than (6921386715),
partition o_ordk1576 values less than (6925781247),
partition o_ordk1577 values less than (6930175778),
partition o_ordk1578 values less than (6934570309),
partition o_ordk1579 values less than (6938964840),
partition o_ordk1580 values less than (6943359372),
partition o_ordk1581 values less than (6947753903),
partition o_ordk1582 values less than (6952148434),
partition o_ordk1583 values less than (6956542965),
partition o_ordk1584 values less than (6960937497),
partition o_ordk1585 values less than (6965332028),
partition o_ordk1586 values less than (6969726559),
partition o_ordk1587 values less than (6974121090),
partition o_ordk1588 values less than (6978515622),
partition o_ordk1589 values less than (6982910153),
partition o_ordk1590 values less than (6987304684),
partition o_ordk1591 values less than (6991699215),
partition o_ordk1592 values less than (6996093747),
partition o_ordk1593 values less than (7000488278),
partition o_ordk1594 values less than (7004882809),
partition o_ordk1595 values less than (7009277340),
partition o_ordk1596 values less than (7013671871),
partition o_ordk1597 values less than (7018066403),
partition o_ordk1598 values less than (7022460934),
partition o_ordk1599 values less than (7026855465),
partition o_ordk1600 values less than (7031249996),
partition o_ordk1601 values less than (7035644528),
partition o_ordk1602 values less than (7040039059),
partition o_ordk1603 values less than (7044433590),
partition o_ordk1604 values less than (7048828121),
partition o_ordk1605 values less than (7053222653),
partition o_ordk1606 values less than (7057617184),
partition o_ordk1607 values less than (7062011715),
partition o_ordk1608 values less than (7066406246),
partition o_ordk1609 values less than (7070800778),
partition o_ordk1610 values less than (7075195309),
partition o_ordk1611 values less than (7079589840),
partition o_ordk1612 values less than (7083984371),
partition o_ordk1613 values less than (7088378903),
partition o_ordk1614 values less than (7092773434),
partition o_ordk1615 values less than (7097167965),
partition o_ordk1616 values less than (7101562496),
partition o_ordk1617 values less than (7105957028),
partition o_ordk1618 values less than (7110351559),
partition o_ordk1619 values less than (7114746090),
partition o_ordk1620 values less than (7119140621),
partition o_ordk1621 values less than (7123535153),
partition o_ordk1622 values less than (7127929684),
partition o_ordk1623 values less than (7132324215),
partition o_ordk1624 values less than (7136718746),
partition o_ordk1625 values less than (7141113278),
partition o_ordk1626 values less than (7145507809),
partition o_ordk1627 values less than (7149902340),
partition o_ordk1628 values less than (7154296871),
partition o_ordk1629 values less than (7158691403),
partition o_ordk1630 values less than (7163085934),
partition o_ordk1631 values less than (7167480465),
partition o_ordk1632 values less than (7171874996),
partition o_ordk1633 values less than (7176269528),
partition o_ordk1634 values less than (7180664059),
partition o_ordk1635 values less than (7185058590),
partition o_ordk1636 values less than (7189453121),
partition o_ordk1637 values less than (7193847653),
partition o_ordk1638 values less than (7198242184),
partition o_ordk1639 values less than (7202636715),
partition o_ordk1640 values less than (7207031246),
partition o_ordk1641 values less than (7211425778),
partition o_ordk1642 values less than (7215820309),
partition o_ordk1643 values less than (7220214840),
partition o_ordk1644 values less than (7224609371),
partition o_ordk1645 values less than (7229003903),
partition o_ordk1646 values less than (7233398434),
partition o_ordk1647 values less than (7237792965),
partition o_ordk1648 values less than (7242187496),
partition o_ordk1649 values less than (7246582028),
partition o_ordk1650 values less than (7250976559),
partition o_ordk1651 values less than (7255371090),
partition o_ordk1652 values less than (7259765621),
partition o_ordk1653 values less than (7264160153),
partition o_ordk1654 values less than (7268554684),
partition o_ordk1655 values less than (7272949215),
partition o_ordk1656 values less than (7277343746),
partition o_ordk1657 values less than (7281738278),
partition o_ordk1658 values less than (7286132809),
partition o_ordk1659 values less than (7290527340),
partition o_ordk1660 values less than (7294921871),
partition o_ordk1661 values less than (7299316403),
partition o_ordk1662 values less than (7303710934),
partition o_ordk1663 values less than (7308105465),
partition o_ordk1664 values less than (7312499996),
partition o_ordk1665 values less than (7316894528),
partition o_ordk1666 values less than (7321289059),
partition o_ordk1667 values less than (7325683590),
partition o_ordk1668 values less than (7330078121),
partition o_ordk1669 values less than (7334472653),
partition o_ordk1670 values less than (7338867184),
partition o_ordk1671 values less than (7343261715),
partition o_ordk1672 values less than (7347656246),
partition o_ordk1673 values less than (7352050778),
partition o_ordk1674 values less than (7356445309),
partition o_ordk1675 values less than (7360839840),
partition o_ordk1676 values less than (7365234371),
partition o_ordk1677 values less than (7369628903),
partition o_ordk1678 values less than (7374023434),
partition o_ordk1679 values less than (7378417965),
partition o_ordk1680 values less than (7382812496),
partition o_ordk1681 values less than (7387207028),
partition o_ordk1682 values less than (7391601559),
```

```
partition o_ordk1683 values less than (7395996090),
partition o_ordk1684 values less than (7400390621),
partition o_ordk1685 values less than (7404785153),
partition o_ordk1686 values less than (7409179684),
partition o_ordk1687 values less than (7413574215),
partition o_ordk1688 values less than (7417968746),
partition o_ordk1689 values less than (7422363278),
partition o_ordk1690 values less than (7426757809),
partition o_ordk1691 values less than (7431152340),
partition o_ordk1692 values less than (7435546871),
partition o_ordk1693 values less than (7439941403),
partition o_ordk1694 values less than (7444335934),
partition o_ordk1695 values less than (7448730465),
partition o_ordk1696 values less than (7453124996),
partition o_ordk1697 values less than (7457519528),
partition o_ordk1698 values less than (7461914059),
partition o_ordk1699 values less than (7466308590),
partition o_ordk1700 values less than (7470703121),
partition o_ordk1701 values less than (7475097653),
partition o_ordk1702 values less than (7479492184),
partition o_ordk1703 values less than (7483886715),
partition o_ordk1704 values less than (7488281246),
partition o_ordk1705 values less than (7492675778),
partition o_ordk1706 values less than (7497070309),
partition o_ordk1707 values less than (7501464840),
partition o_ordk1708 values less than (7505859371),
partition o_ordk1709 values less than (7510253902),
partition o_ordk1710 values less than (7514648434),
partition o_ordk1711 values less than (7519042965),
partition o_ordk1712 values less than (7523437496),
partition o_ordk1713 values less than (7527832027),
partition o_ordk1714 values less than (7532226559),
partition o_ordk1715 values less than (7536621090),
partition o_ordk1716 values less than (7541015621),
partition o_ordk1717 values less than (7545410152),
partition o_ordk1718 values less than (7549804684),
partition o_ordk1719 values less than (7554199215),
partition o_ordk1720 values less than (7558593746),
partition o_ordk1721 values less than (7562988277),
partition o_ordk1722 values less than (7567382809),
partition o_ordk1723 values less than (7571777340),
partition o_ordk1724 values less than (7576171871),
partition o_ordk1725 values less than (7580566402),
partition o_ordk1726 values less than (7584960934),
partition o_ordk1727 values less than (7589355465),
partition o_ordk1728 values less than (7593749996),
partition o_ordk1729 values less than (7598144527),
partition o_ordk1730 values less than (7602539059),
partition o_ordk1731 values less than (7606933590),
partition o_ordk1732 values less than (7611328121),
partition o_ordk1733 values less than (7615722652),
partition o_ordk1734 values less than (7620117184),
partition o_ordk1735 values less than (7624511715),
partition o_ordk1736 values less than (7628906246),
partition o_ordk1737 values less than (7633300777),
partition o_ordk1738 values less than (7637695309),
partition o_ordk1739 values less than (7642089840),
partition o_ordk1740 values less than (7646484371),
partition o_ordk1741 values less than (7650878902),
partition o_ordk1742 values less than (7655273434),
partition o_ordk1743 values less than (7659667965),
partition o_ordk1744 values less than (7664062496),
partition o_ordk1745 values less than (7668457027),
partition o_ordk1746 values less than (7672851559),
partition o_ordk1747 values less than (7677246090),
partition o_ordk1748 values less than (7681640621),
partition o_ordk1749 values less than (7686035152),
partition o_ordk1750 values less than (7690429684),
partition o_ordk1751 values less than (7694824215),
partition o_ordk1752 values less than (7699218746),
partition o_ordk1753 values less than (7703613277),
partition o_ordk1754 values less than (7708007809),
partition o_ordk1755 values less than (7712402340),
partition o_ordk1756 values less than (7716796871),
partition o_ordk1757 values less than (7721191402),
partition o_ordk1758 values less than (7725585934),
partition o_ordk1759 values less than (7729980465),
partition o_ordk1760 values less than (7734374996),
partition o_ordk1761 values less than (7738769527),
partition o_ordk1762 values less than (7743164059),
partition o_ordk1763 values less than (7747558590),
partition o_ordk1764 values less than (7751953121),
partition o_ordk1765 values less than (7756347652),
partition o_ordk1766 values less than (7760742184),
partition o_ordk1767 values less than (7765136715),
partition o_ordk1768 values less than (7769531246),
partition o_ordk1769 values less than (7773925777),
partition o_ordk1770 values less than (7778320309),
partition o_ordk1771 values less than (7782714840),
partition o_ordk1772 values less than (7787109371),
partition o_ordk1773 values less than (7791503902),
partition o_ordk1774 values less than (7795898434),
partition o_ordk1775 values less than (7800292965),
partition o_ordk1776 values less than (7804687496),
partition o_ordk1777 values less than (7809082027),
partition o_ordk1778 values less than (7813476559),
partition o_ordk1779 values less than (7817871090),
partition o_ordk1780 values less than (7822265621),
partition o_ordk1781 values less than (7826660152),
partition o_ordk1782 values less than (7831054684),
partition o_ordk1783 values less than (7835449215),
partition o_ordk1784 values less than (7839843746),
partition o_ordk1785 values less than (7844238277),
partition o_ordk1786 values less than (7848632809),
partition o_ordk1787 values less than (7853027340),
partition o_ordk1788 values less than (7857421871),
partition o_ordk1789 values less than (7861816402),
partition o_ordk1790 values less than (7866210934),
partition o_ordk1791 values less than (7870605465),
partition o_ordk1792 values less than (7874999996),
partition o_ordk1793 values less than (7879394527),
partition o_ordk1794 values less than (7883789059),
partition o_ordk1795 values less than (7888183590),
partition o_ordk1796 values less than (7892578121),
partition o_ordk1797 values less than (7896972652),
partition o_ordk1798 values less than (7901367184),
partition o_ordk1799 values less than (7905761715),
partition o_ordk1800 values less than (7910156246),
partition o_ordk1801 values less than (7914550777),
partition o_ordk1802 values less than (7918945309),
partition o_ordk1803 values less than (7923339840),
partition o_ordk1804 values less than (7927734371),
partition o_ordk1805 values less than (7932128902),
partition o_ordk1806 values less than (7936523434),
partition o_ordk1807 values less than (7940917965),
partition o_ordk1808 values less than (7945312496),
partition o_ordk1809 values less than (7949707027),
partition o_ordk1810 values less than (7954101559),
partition o_ordk1811 values less than (7958496090),
partition o_ordk1812 values less than (7962890621),
partition o_ordk1813 values less than (7967285152),
partition o_ordk1814 values less than (7971679684),
partition o_ordk1815 values less than (7976074215),
partition o_ordk1816 values less than (7980468746),
partition o_ordk1817 values less than (7984863277),
partition o_ordk1818 values less than (7989257809),
partition o_ordk1819 values less than (7993652340),
partition o_ordk1820 values less than (7998046871),
partition o_ordk1821 values less than (8002441402),
partition o_ordk1822 values less than (8006835933),
partition o_ordk1823 values less than (8011230465),
partition o_ordk1824 values less than (8015624996),
partition o_ordk1825 values less than (8020019527),
partition o_ordk1826 values less than (8024414058),
partition o_ordk1827 values less than (8028808590),
partition o_ordk1828 values less than (8033203121),
partition o_ordk1829 values less than (8037597652),
partition o_ordk1830 values less than (8041992183),
partition o_ordk1831 values less than (8046386715),
partition o_ordk1832 values less than (8050781246),
partition o_ordk1833 values less than (8055175777),
partition o_ordk1834 values less than (8059570308),
partition o_ordk1835 values less than (8063964840),
partition o_ordk1836 values less than (8068359371),
partition o_ordk1837 values less than (8072753902),
partition o_ordk1838 values less than (8077148433),
partition o_ordk1839 values less than (8081542965),
partition o_ordk1840 values less than (8085937496),
partition o_ordk1841 values less than (8090332027),
partition o_ordk1842 values less than (8094726558),
partition o_ordk1843 values less than (8099121090),
partition o_ordk1844 values less than (8103515621),
partition o_ordk1845 values less than (8107910152),
partition o_ordk1846 values less than (8112304683),
partition o_ordk1847 values less than (8116699215),
partition o_ordk1848 values less than (8121093746),
partition o_ordk1849 values less than (8125488277),
partition o_ordk1850 values less than (8129882808),
partition o_ordk1851 values less than (8134277340),
partition o_ordk1852 values less than (8138671871),
partition o_ordk1853 values less than (8143066402),
partition o_ordk1854 values less than (8147460933),
partition o_ordk1855 values less than (8151855465),
partition o_ordk1856 values less than (8156249996),
partition o_ordk1857 values less than (8160644527),
partition o_ordk1858 values less than (8165039058),
partition o_ordk1859 values less than (8169433590),
partition o_ordk1860 values less than (8173828121),
partition o_ordk1861 values less than (8178222652),
partition o_ordk1862 values less than (8182617183),
```

```
partition o_ordk1863 values less than (8187011715),
partition o_ordk1864 values less than (8191406246),
partition o_ordk1865 values less than (8195800777),
partition o_ordk1866 values less than (8200195308),
partition o_ordk1867 values less than (8204589840),
partition o_ordk1868 values less than (8208984371),
partition o_ordk1869 values less than (8213378902),
partition o_ordk1870 values less than (8217773433),
partition o_ordk1871 values less than (8222167965),
partition o_ordk1872 values less than (8226562496),
partition o_ordk1873 values less than (8230957027),
partition o_ordk1874 values less than (8235351558),
partition o_ordk1875 values less than (8239746090),
partition o_ordk1876 values less than (8244140621),
partition o_ordk1877 values less than (8248535152),
partition o_ordk1878 values less than (8252929683),
partition o_ordk1879 values less than (8257324215),
partition o_ordk1880 values less than (8261718746),
partition o_ordk1881 values less than (8266113277),
partition o_ordk1882 values less than (8270507808),
partition o_ordk1883 values less than (8274902340),
partition o_ordk1884 values less than (8279296871),
partition o_ordk1885 values less than (8283691402),
partition o_ordk1886 values less than (8288085933),
partition o_ordk1887 values less than (8292480465),
partition o_ordk1888 values less than (8296874996),
partition o_ordk1889 values less than (8301269527),
partition o_ordk1890 values less than (8305664058),
partition o_ordk1891 values less than (8310058590),
partition o_ordk1892 values less than (8314453121),
partition o_ordk1893 values less than (8318847652),
partition o_ordk1894 values less than (8323242183),
partition o_ordk1895 values less than (8327636715),
partition o_ordk1896 values less than (8332031246),
partition o_ordk1897 values less than (8336425777),
partition o_ordk1898 values less than (8340820308),
partition o_ordk1899 values less than (8345214840),
partition o_ordk1900 values less than (8349609371),
partition o_ordk1901 values less than (8354003902),
partition o_ordk1902 values less than (8358398433),
partition o_ordk1903 values less than (8362792965),
partition o_ordk1904 values less than (8367187496),
partition o_ordk1905 values less than (8371582027),
partition o_ordk1906 values less than (8375976558),
partition o_ordk1907 values less than (8380371090),
partition o_ordk1908 values less than (8384765621),
partition o_ordk1909 values less than (8389160152),
partition o_ordk1910 values less than (8393554683),
partition o_ordk1911 values less than (8397949215),
partition o_ordk1912 values less than (8402343746),
partition o_ordk1913 values less than (8406738277),
partition o_ordk1914 values less than (8411132808),
partition o_ordk1915 values less than (8415527340),
partition o_ordk1916 values less than (8419921871),
partition o_ordk1917 values less than (8424316402),
partition o_ordk1918 values less than (8428710933),
partition o_ordk1919 values less than (8433105465),
partition o_ordk1920 values less than (8437499996),
partition o_ordk1921 values less than (8441894527),
partition o_ordk1922 values less than (8446289058),
partition o_ordk1923 values less than (8450683590),
partition o_ordk1924 values less than (8455078121),
partition o_ordk1925 values less than (8459472652),
partition o_ordk1926 values less than (8463867183),
partition o_ordk1927 values less than (8468261715),
partition o_ordk1928 values less than (8472656246),
partition o_ordk1929 values less than (8477050777),
partition o_ordk1930 values less than (8481445308),
partition o_ordk1931 values less than (8485839840),
partition o_ordk1932 values less than (8490234371),
partition o_ordk1933 values less than (8494628902),
partition o_ordk1934 values less than (8499023433),
partition o_ordk1935 values less than (8503417964),
partition o_ordk1936 values less than (8507812496),
partition o_ordk1937 values less than (8512207027),
partition o_ordk1938 values less than (8516601558),
partition o_ordk1939 values less than (8520996089),
partition o_ordk1940 values less than (8525390621),
partition o_ordk1941 values less than (8529785152),
partition o_ordk1942 values less than (8534179683),
partition o_ordk1943 values less than (8538574214),
partition o_ordk1944 values less than (8542968746),
partition o_ordk1945 values less than (8547363277),
partition o_ordk1946 values less than (8551757808),
partition o_ordk1947 values less than (8556152339),
partition o_ordk1948 values less than (8560546871),
partition o_ordk1949 values less than (8564941402),
partition o_ordk1950 values less than (8569335933),
partition o_ordk1951 values less than (8573730464),
partition o_ordk1952 values less than (8578124996),
partition o_ordk1953 values less than (8582519527),
partition o_ordk1954 values less than (8586914058),
partition o_ordk1955 values less than (8591308589),
partition o_ordk1956 values less than (8595703121),
partition o_ordk1957 values less than (8600097652),
partition o_ordk1958 values less than (8604492183),
partition o_ordk1959 values less than (8608886714),
partition o_ordk1960 values less than (8613281246),
partition o_ordk1961 values less than (8617675777),
partition o_ordk1962 values less than (8622070308),
partition o_ordk1963 values less than (8626464839),
partition o_ordk1964 values less than (8630859371),
partition o_ordk1965 values less than (8635253902),
partition o_ordk1966 values less than (8639648433),
partition o_ordk1967 values less than (8644042964),
partition o_ordk1968 values less than (8648437496),
partition o_ordk1969 values less than (8652832027),
partition o_ordk1970 values less than (8657226558),
partition o_ordk1971 values less than (8661621089),
partition o_ordk1972 values less than (8666015621),
partition o_ordk1973 values less than (8670410152),
partition o_ordk1974 values less than (8674804683),
partition o_ordk1975 values less than (8679199214),
partition o_ordk1976 values less than (8683593746),
partition o_ordk1977 values less than (8687988277),
partition o_ordk1978 values less than (8692382808),
partition o_ordk1979 values less than (8696777339),
partition o_ordk1980 values less than (8701171871),
partition o_ordk1981 values less than (8705566402),
partition o_ordk1982 values less than (8709960933),
partition o_ordk1983 values less than (8714355464),
partition o_ordk1984 values less than (8718749996),
partition o_ordk1985 values less than (8723144527),
partition o_ordk1986 values less than (8727539058),
partition o_ordk1987 values less than (8731933589),
partition o_ordk1988 values less than (8736328121),
partition o_ordk1989 values less than (8740722652),
partition o_ordk1990 values less than (8745117183),
partition o_ordk1991 values less than (8749511714),
partition o_ordk1992 values less than (8753906246),
partition o_ordk1993 values less than (8758300777),
partition o_ordk1994 values less than (8762695308),
partition o_ordk1995 values less than (8767089839),
partition o_ordk1996 values less than (8771484371),
partition o_ordk1997 values less than (8775878902),
partition o_ordk1998 values less than (8780273433),
partition o_ordk1999 values less than (8784667964),
partition o_ordk2000 values less than (8789062496),
partition o_ordk2001 values less than (8793457027),
partition o_ordk2002 values less than (8797851558),
partition o_ordk2003 values less than (8802246089),
partition o_ordk2004 values less than (8806640621),
partition o_ordk2005 values less than (8811035152),
partition o_ordk2006 values less than (8815429683),
partition o_ordk2007 values less than (8819824214),
partition o_ordk2008 values less than (8824218746),
partition o_ordk2009 values less than (8828613277),
partition o_ordk2010 values less than (8833007808),
partition o_ordk2011 values less than (8837402339),
partition o_ordk2012 values less than (8841796871),
partition o_ordk2013 values less than (8846191402),
partition o_ordk2014 values less than (8850585933),
partition o_ordk2015 values less than (8854980464),
partition o_ordk2016 values less than (8859374996),
partition o_ordk2017 values less than (8863769527),
partition o_ordk2018 values less than (8868164058),
partition o_ordk2019 values less than (8872558589),
partition o_ordk2020 values less than (8876953121),
partition o_ordk2021 values less than (8881347652),
partition o_ordk2022 values less than (8885742183),
partition o_ordk2023 values less than (8890136714),
partition o_ordk2024 values less than (8894531246),
partition o_ordk2025 values less than (8898925777),
partition o_ordk2026 values less than (8903320308),
partition o_ordk2027 values less than (8907714839),
partition o_ordk2028 values less than (8912109371),
partition o_ordk2029 values less than (8916503902),
partition o_ordk2030 values less than (8920898433),
partition o_ordk2031 values less than (8925292964),
partition o_ordk2032 values less than (8929687496),
partition o_ordk2033 values less than (8934082027),
partition o_ordk2034 values less than (8938476558),
partition o_ordk2035 values less than (8942871089),
partition o_ordk2036 values less than (8947265621),
partition o_ordk2037 values less than (8951660152),
partition o_ordk2038 values less than (8956054683),
partition o_ordk2039 values less than (8960449214),
partition o_ordk2040 values less than (8964843746),
partition o_ordk2041 values less than (8969238277),
partition o_ordk2042 values less than (8973632808),
```

```
partition o_ordk2043 values less than (8978027339),
partition o_ordk2044 values less than (8982421871),
partition o_ordk2045 values less than (8986816402),
partition o_ordk2046 values less than (8991210933),
partition o_ordk2047 values less than (8995605464),
partition o_ordk2048 values less than (8999999996),
partition o_ordk2049 values less than (9004394527),
partition o_ordk2050 values less than (9008789058),
partition o_ordk2051 values less than (9013183589),
partition o_ordk2052 values less than (9017578120),
partition o_ordk2053 values less than (9021972652),
partition o_ordk2054 values less than (9026367183),
partition o_ordk2055 values less than (9030761714),
partition o_ordk2056 values less than (9035156245),
partition o_ordk2057 values less than (9039550777),
partition o_ordk2058 values less than (9043945308),
partition o_ordk2059 values less than (9048339839),
partition o_ordk2060 values less than (9052734370),
partition o_ordk2061 values less than (9057128902),
partition o_ordk2062 values less than (9061523433),
partition o_ordk2063 values less than (9065917964),
partition o_ordk2064 values less than (9070312495),
partition o_ordk2065 values less than (9074707027),
partition o_ordk2066 values less than (9079101558),
partition o_ordk2067 values less than (9083496089),
partition o_ordk2068 values less than (9087890620),
partition o_ordk2069 values less than (9092285152),
partition o_ordk2070 values less than (9096679683),
partition o_ordk2071 values less than (9101074214),
partition o_ordk2072 values less than (9105468745),
partition o_ordk2073 values less than (9109863277),
partition o_ordk2074 values less than (9114257808),
partition o_ordk2075 values less than (9118652339),
partition o_ordk2076 values less than (9123046870),
partition o_ordk2077 values less than (9127441402),
partition o_ordk2078 values less than (9131835933),
partition o_ordk2079 values less than (9136230464),
partition o_ordk2080 values less than (9140624995),
partition o_ordk2081 values less than (9145019527),
partition o_ordk2082 values less than (9149414058),
partition o_ordk2083 values less than (9153808589),
partition o_ordk2084 values less than (9158203120),
partition o_ordk2085 values less than (9162597652),
partition o_ordk2086 values less than (9166992183),
partition o_ordk2087 values less than (9171386714),
partition o_ordk2088 values less than (9175781245),
partition o_ordk2089 values less than (9180175777),
partition o_ordk2090 values less than (9184570308),
partition o_ordk2091 values less than (9188964839),
partition o_ordk2092 values less than (9193359370),
partition o_ordk2093 values less than (9197753902),
partition o_ordk2094 values less than (9202148433),
partition o_ordk2095 values less than (9206542964),
partition o_ordk2096 values less than (9210937495),
partition o_ordk2097 values less than (9215332027),
partition o_ordk2098 values less than (9219726558),
partition o_ordk2099 values less than (9224121089),
partition o_ordk2100 values less than (9228515620),
partition o_ordk2101 values less than (9232910152),
partition o_ordk2102 values less than (9237304683),
partition o_ordk2103 values less than (9241699214),
partition o_ordk2104 values less than (9246093745),
partition o_ordk2105 values less than (9250488277),
partition o_ordk2106 values less than (9254882808),
partition o_ordk2107 values less than (9259277339),
partition o_ordk2108 values less than (9263671870),
partition o_ordk2109 values less than (9268066402),
partition o_ordk2110 values less than (9272460933),
partition o_ordk2111 values less than (9276855464),
partition o_ordk2112 values less than (9281249995),
partition o_ordk2113 values less than (9285644527),
partition o_ordk2114 values less than (9290039058),
partition o_ordk2115 values less than (9294433589),
partition o_ordk2116 values less than (9298828120),
partition o_ordk2117 values less than (9303222652),
partition o_ordk2118 values less than (9307617183),
partition o_ordk2119 values less than (9312011714),
partition o_ordk2120 values less than (9316406245),
partition o_ordk2121 values less than (9320800777),
partition o_ordk2122 values less than (9325195308),
partition o_ordk2123 values less than (9329589839),
partition o_ordk2124 values less than (9333984370),
partition o_ordk2125 values less than (9338378902),
partition o_ordk2126 values less than (9342773433),
partition o_ordk2127 values less than (9347167964),
partition o_ordk2128 values less than (9351562495),
partition o_ordk2129 values less than (9355957027),
partition o_ordk2130 values less than (9360351558),
partition o_ordk2131 values less than (9364746089),
partition o_ordk2132 values less than (9369140620),
partition o_ordk2133 values less than (9373535152),
partition o_ordk2134 values less than (9377929683),
partition o_ordk2135 values less than (9382324214),
partition o_ordk2136 values less than (9386718745),
partition o_ordk2137 values less than (9391113277),
partition o_ordk2138 values less than (9395507808),
partition o_ordk2139 values less than (9399902339),
partition o_ordk2140 values less than (9404296870),
partition o_ordk2141 values less than (9408691402),
partition o_ordk2142 values less than (9413085933),
partition o_ordk2143 values less than (9417480464),
partition o_ordk2144 values less than (9421874995),
partition o_ordk2145 values less than (9426269527),
partition o_ordk2146 values less than (9430664058),
partition o_ordk2147 values less than (9435058589),
partition o_ordk2148 values less than (9439453120),
partition o_ordk2149 values less than (9443847652),
partition o_ordk2150 values less than (9448242183),
partition o_ordk2151 values less than (9452636714),
partition o_ordk2152 values less than (9457031245),
partition o_ordk2153 values less than (9461425777),
partition o_ordk2154 values less than (9465820308),
partition o_ordk2155 values less than (9470214839),
partition o_ordk2156 values less than (9474609370),
partition o_ordk2157 values less than (9479003902),
partition o_ordk2158 values less than (9483398433),
partition o_ordk2159 values less than (9487792964),
partition o_ordk2160 values less than (9492187495),
partition o_ordk2161 values less than (9496582027),
partition o_ordk2162 values less than (9500976558),
partition o_ordk2163 values less than (9505371089),
partition o_ordk2164 values less than (9509765620),
partition o_ordk2165 values less than (9514160151),
partition o_ordk2166 values less than (9518554683),
partition o_ordk2167 values less than (9522949214),
partition o_ordk2168 values less than (9527343745),
partition o_ordk2169 values less than (9531738276),
partition o_ordk2170 values less than (9536132808),
partition o_ordk2171 values less than (9540527339),
partition o_ordk2172 values less than (9544921870),
partition o_ordk2173 values less than (9549316401),
partition o_ordk2174 values less than (9553710933),
partition o_ordk2175 values less than (9558105464),
partition o_ordk2176 values less than (9562499995),
partition o_ordk2177 values less than (9566894526),
partition o_ordk2178 values less than (9571289058),
partition o_ordk2179 values less than (9575683589),
partition o_ordk2180 values less than (9580078120),
partition o_ordk2181 values less than (9584472651),
partition o_ordk2182 values less than (9588867183),
partition o_ordk2183 values less than (9593261714),
partition o_ordk2184 values less than (9597656245),
partition o_ordk2185 values less than (9602050776),
partition o_ordk2186 values less than (9606445308),
partition o_ordk2187 values less than (9610839839),
partition o_ordk2188 values less than (9615234370),
partition o_ordk2189 values less than (9619628901),
partition o_ordk2190 values less than (9624023433),
partition o_ordk2191 values less than (9628417964),
partition o_ordk2192 values less than (9632812495),
partition o_ordk2193 values less than (9637207026),
partition o_ordk2194 values less than (9641601558),
partition o_ordk2195 values less than (9645996089),
partition o_ordk2196 values less than (9650390620),
partition o_ordk2197 values less than (9654785151),
partition o_ordk2198 values less than (9659179683),
partition o_ordk2199 values less than (9663574214),
partition o_ordk2200 values less than (9667968745),
partition o_ordk2201 values less than (9672363276),
partition o_ordk2202 values less than (9676757808),
partition o_ordk2203 values less than (9681152339),
partition o_ordk2204 values less than (9685546870),
partition o_ordk2205 values less than (9689941401),
partition o_ordk2206 values less than (9694335933),
partition o_ordk2207 values less than (9698730464),
partition o_ordk2208 values less than (9703124995),
partition o_ordk2209 values less than (9707519526),
partition o_ordk2210 values less than (9711914058),
partition o_ordk2211 values less than (9716308589),
partition o_ordk2212 values less than (9720703120),
partition o_ordk2213 values less than (9725097651),
partition o_ordk2214 values less than (9729492183),
partition o_ordk2215 values less than (9733886714),
partition o_ordk2216 values less than (9738281245),
partition o_ordk2217 values less than (9742675776),
partition o_ordk2218 values less than (9747070308),
partition o_ordk2219 values less than (9751464839),
partition o_ordk2220 values less than (9755859370),
partition o_ordk2221 values less than (9760253901),
partition o_ordk2222 values less than (9764648433),
```

```
partition o_ordk2223 values less than (9769042964),
partition o_ordk2224 values less than (9773437495),
partition o_ordk2225 values less than (9777832026),
partition o_ordk2226 values less than (9782226558),
partition o_ordk2227 values less than (9786621089),
partition o_ordk2228 values less than (9791015620),
partition o_ordk2229 values less than (9795410151),
partition o_ordk2230 values less than (9799804683),
partition o_ordk2231 values less than (9804199214),
partition o_ordk2232 values less than (9808593745),
partition o_ordk2233 values less than (9812988276),
partition o_ordk2234 values less than (9817382808),
partition o_ordk2235 values less than (9821777339),
partition o_ordk2236 values less than (9826171870),
partition o_ordk2237 values less than (9830566401),
partition o_ordk2238 values less than (9834960933),
partition o_ordk2239 values less than (9839355464),
partition o_ordk2240 values less than (9843749995),
partition o_ordk2241 values less than (9848144526),
partition o_ordk2242 values less than (9852539058),
partition o_ordk2243 values less than (9856933589),
partition o_ordk2244 values less than (9861328120),
partition o_ordk2245 values less than (9865722651),
partition o_ordk2246 values less than (9870117183),
partition o_ordk2247 values less than (9874511714),
partition o_ordk2248 values less than (9878906245),
partition o_ordk2249 values less than (9883300776),
partition o_ordk2250 values less than (9887695308),
partition o_ordk2251 values less than (9892089839),
partition o_ordk2252 values less than (9896484370),
partition o_ordk2253 values less than (9900878901),
partition o_ordk2254 values less than (9905273433),
partition o_ordk2255 values less than (9909667964),
partition o_ordk2256 values less than (9914062495),
partition o_ordk2257 values less than (9918457026),
partition o_ordk2258 values less than (9922851558),
partition o_ordk2259 values less than (9927246089),
partition o_ordk2260 values less than (9931640620),
partition o_ordk2261 values less than (9936035151),
partition o_ordk2262 values less than (9940429683),
partition o_ordk2263 values less than (9944824214),
partition o_ordk2264 values less than (9949218745),
partition o_ordk2265 values less than (9953613276),
partition o_ordk2266 values less than (9958007808),
partition o_ordk2267 values less than (9962402339),
partition o_ordk2268 values less than (9966796870),
partition o_ordk2269 values less than (9971191401),
partition o_ordk2270 values less than (9975585933),
partition o_ordk2271 values less than (9979980464),
partition o_ordk2272 values less than (9984374995),
partition o_ordk2273 values less than (9988769526),
partition o_ordk2274 values less than (9993164058),
partition o_ordk2275 values less than (9997558589),
partition o_ordk2276 values less than (10001953120),
partition o_ordk2277 values less than (10006347651),
partition o_ordk2278 values less than (10010742182),
partition o_ordk2279 values less than (10015136714),
partition o_ordk2280 values less than (10019531245),
partition o_ordk2281 values less than (10023925776),
partition o_ordk2282 values less than (10028320307),
partition o_ordk2283 values less than (10032714839),
partition o_ordk2284 values less than (10037109370),
partition o_ordk2285 values less than (10041503901),
partition o_ordk2286 values less than (10045898432),
partition o_ordk2287 values less than (10050292964),
partition o_ordk2288 values less than (10054687495),
partition o_ordk2289 values less than (10059082026),
partition o_ordk2290 values less than (10063476557),
partition o_ordk2291 values less than (10067871089),
partition o_ordk2292 values less than (10072265620),
partition o_ordk2293 values less than (10076660151),
partition o_ordk2294 values less than (10081054682),
partition o_ordk2295 values less than (10085449214),
partition o_ordk2296 values less than (10089843745),
partition o_ordk2297 values less than (10094238276),
partition o_ordk2298 values less than (10098632807),
partition o_ordk2299 values less than (10103027339),
partition o_ordk2300 values less than (10107421870),
partition o_ordk2301 values less than (10111816401),
partition o_ordk2302 values less than (10116210932),
partition o_ordk2303 values less than (10120605464),
partition o_ordk2304 values less than (10124999995),
partition o_ordk2305 values less than (10129394526),
partition o_ordk2306 values less than (10133789057),
partition o_ordk2307 values less than (10138183589),
partition o_ordk2308 values less than (10142578120),
partition o_ordk2309 values less than (10146972651),
partition o_ordk2310 values less than (10151367182),
partition o_ordk2311 values less than (10155761714),
partition o_ordk2312 values less than (10160156245),
partition o_ordk2313 values less than (10164550776),
partition o_ordk2314 values less than (10168945307),
partition o_ordk2315 values less than (10173339839),
partition o_ordk2316 values less than (10177734370),
partition o_ordk2317 values less than (10182128901),
partition o_ordk2318 values less than (10186523432),
partition o_ordk2319 values less than (10190917964),
partition o_ordk2320 values less than (10195312495),
partition o_ordk2321 values less than (10199707026),
partition o_ordk2322 values less than (10204101557),
partition o_ordk2323 values less than (10208496089),
partition o_ordk2324 values less than (10212890620),
partition o_ordk2325 values less than (10217285151),
partition o_ordk2326 values less than (10221679682),
partition o_ordk2327 values less than (10226074214),
partition o_ordk2328 values less than (10230468745),
partition o_ordk2329 values less than (10234863276),
partition o_ordk2330 values less than (10239257807),
partition o_ordk2331 values less than (10243652339),
partition o_ordk2332 values less than (10248046870),
partition o_ordk2333 values less than (10252441401),
partition o_ordk2334 values less than (10256835932),
partition o_ordk2335 values less than (10261230464),
partition o_ordk2336 values less than (10265624995),
partition o_ordk2337 values less than (10270019526),
partition o_ordk2338 values less than (10274414057),
partition o_ordk2339 values less than (10278808589),
partition o_ordk2340 values less than (10283203120),
partition o_ordk2341 values less than (10287597651),
partition o_ordk2342 values less than (10291992182),
partition o_ordk2343 values less than (10296386714),
partition o_ordk2344 values less than (10300781245),
partition o_ordk2345 values less than (10305175776),
partition o_ordk2346 values less than (10309570307),
partition o_ordk2347 values less than (10313964839),
partition o_ordk2348 values less than (10318359370),
partition o_ordk2349 values less than (10322753901),
partition o_ordk2350 values less than (10327148432),
partition o_ordk2351 values less than (10331542964),
partition o_ordk2352 values less than (10335937495),
partition o_ordk2353 values less than (10340332026),
partition o_ordk2354 values less than (10344726557),
partition o_ordk2355 values less than (10349121089),
partition o_ordk2356 values less than (10353515620),
partition o_ordk2357 values less than (10357910151),
partition o_ordk2358 values less than (10362304682),
partition o_ordk2359 values less than (10366699214),
partition o_ordk2360 values less than (10371093745),
partition o_ordk2361 values less than (10375488276),
partition o_ordk2362 values less than (10379882807),
partition o_ordk2363 values less than (10384277339),
partition o_ordk2364 values less than (10388671870),
partition o_ordk2365 values less than (10393066401),
partition o_ordk2366 values less than (10397460932),
partition o_ordk2367 values less than (10401855464),
partition o_ordk2368 values less than (10406249995),
partition o_ordk2369 values less than (10410644526),
partition o_ordk2370 values less than (10415039057),
partition o_ordk2371 values less than (10419433589),
partition o_ordk2372 values less than (10423828120),
partition o_ordk2373 values less than (10428222651),
partition o_ordk2374 values less than (10432617182),
partition o_ordk2375 values less than (10437011714),
partition o_ordk2376 values less than (10441406245),
partition o_ordk2377 values less than (10445800776),
partition o_ordk2378 values less than (10450195307),
partition o_ordk2379 values less than (10454589839),
partition o_ordk2380 values less than (10458984370),
partition o_ordk2381 values less than (10463378901),
partition o_ordk2382 values less than (10467773432),
partition o_ordk2383 values less than (10472167964),
partition o_ordk2384 values less than (10476562495),
partition o_ordk2385 values less than (10480957026),
partition o_ordk2386 values less than (10485351557),
partition o_ordk2387 values less than (10489746089),
partition o_ordk2388 values less than (10494140620),
partition o_ordk2389 values less than (10498535151),
partition o_ordk2390 values less than (10502929682),
partition o_ordk2391 values less than (10507324213),
partition o_ordk2392 values less than (10511718745),
partition o_ordk2393 values less than (10516113276),
partition o_ordk2394 values less than (10520507807),
partition o_ordk2395 values less than (10524902338),
partition o_ordk2396 values less than (10529296870),
partition o_ordk2397 values less than (10533691401),
partition o_ordk2398 values less than (10538085932),
partition o_ordk2399 values less than (10542480463),
partition o_ordk2400 values less than (10546874995),
partition o_ordk2401 values less than (10551269526),
partition o_ordk2402 values less than (10555664057),
```

```
partition o_ordk2403 values less than (10560058588),
partition o_ordk2404 values less than (10564453120),
partition o_ordk2405 values less than (10568847651),
partition o_ordk2406 values less than (10573242182),
partition o_ordk2407 values less than (10577636713),
partition o_ordk2408 values less than (10582031245),
partition o_ordk2409 values less than (10586425776),
partition o_ordk2410 values less than (10590820307),
partition o_ordk2411 values less than (10595214838),
partition o_ordk2412 values less than (10599609370),
partition o_ordk2413 values less than (10604003901),
partition o_ordk2414 values less than (10608398432),
partition o_ordk2415 values less than (10612792963),
partition o_ordk2416 values less than (10617187495),
partition o_ordk2417 values less than (10621582026),
partition o_ordk2418 values less than (10625976557),
partition o_ordk2419 values less than (10630371088),
partition o_ordk2420 values less than (10634765620),
partition o_ordk2421 values less than (10639160151),
partition o_ordk2422 values less than (10643554682),
partition o_ordk2423 values less than (10647949213),
partition o_ordk2424 values less than (10652343745),
partition o_ordk2425 values less than (10656738276),
partition o_ordk2426 values less than (10661132807),
partition o_ordk2427 values less than (10665527338),
partition o_ordk2428 values less than (10669921870),
partition o_ordk2429 values less than (10674316401),
partition o_ordk2430 values less than (10678710932),
partition o_ordk2431 values less than (10683105463),
partition o_ordk2432 values less than (10687499995),
partition o_ordk2433 values less than (10691894526),
partition o_ordk2434 values less than (10696289057),
partition o_ordk2435 values less than (10700683588),
partition o_ordk2436 values less than (10705078120),
partition o_ordk2437 values less than (10709472651),
partition o_ordk2438 values less than (10713867182),
partition o_ordk2439 values less than (10718261713),
partition o_ordk2440 values less than (10722656245),
partition o_ordk2441 values less than (10727050776),
partition o_ordk2442 values less than (10731445307),
partition o_ordk2443 values less than (10735839838),
partition o_ordk2444 values less than (10740234370),
partition o_ordk2445 values less than (10744628901),
partition o_ordk2446 values less than (10749023432),
partition o_ordk2447 values less than (10753417963),
partition o_ordk2448 values less than (10757812495),
partition o_ordk2449 values less than (10762207026),
partition o_ordk2450 values less than (10766601557),
partition o_ordk2451 values less than (10770996088),
partition o_ordk2452 values less than (10775390620),
partition o_ordk2453 values less than (10779785151),
partition o_ordk2454 values less than (10784179682),
partition o_ordk2455 values less than (10788574213),
partition o_ordk2456 values less than (10792968745),
partition o_ordk2457 values less than (10797363276),
partition o_ordk2458 values less than (10801757807),
partition o_ordk2459 values less than (10806152338),
partition o_ordk2460 values less than (10810546870),
partition o_ordk2461 values less than (10814941401),
partition o_ordk2462 values less than (10819335932),
partition o_ordk2463 values less than (10823730463),
partition o_ordk2464 values less than (10828124995),
partition o_ordk2465 values less than (10832519526),
partition o_ordk2466 values less than (10836914057),
partition o_ordk2467 values less than (10841308588),
partition o_ordk2468 values less than (10845703120),
partition o_ordk2469 values less than (10850097651),
partition o_ordk2470 values less than (10854492182),
partition o_ordk2471 values less than (10858886713),
partition o_ordk2472 values less than (10863281245),
partition o_ordk2473 values less than (10867675776),
partition o_ordk2474 values less than (10872070307),
partition o_ordk2475 values less than (10876464838),
partition o_ordk2476 values less than (10880859370),
partition o_ordk2477 values less than (10885253901),
partition o_ordk2478 values less than (10889648432),
partition o_ordk2479 values less than (10894042963),
partition o_ordk2480 values less than (10898437495),
partition o_ordk2481 values less than (10902832026),
partition o_ordk2482 values less than (10907226557),
partition o_ordk2483 values less than (10911621088),
partition o_ordk2484 values less than (10916015620),
partition o_ordk2485 values less than (10920410151),
partition o_ordk2486 values less than (10924804682),
partition o_ordk2487 values less than (10929199213),
partition o_ordk2488 values less than (10933593745),
partition o_ordk2489 values less than (10937988276),
partition o_ordk2490 values less than (10942382807),
partition o_ordk2491 values less than (10946777338),
partition o_ordk2492 values less than (10951171870),
partition o_ordk2493 values less than (10955566401),
partition o_ordk2494 values less than (10959960932),
partition o_ordk2495 values less than (10964355463),
partition o_ordk2496 values less than (10968749995),
partition o_ordk2497 values less than (10973144526),
partition o_ordk2498 values less than (10977539057),
partition o_ordk2499 values less than (10981933588),
partition o_ordk2500 values less than (10986328120),
partition o_ordk2501 values less than (10990722651),
partition o_ordk2502 values less than (10995117182),
partition o_ordk2503 values less than (10999511713),
partition o_ordk2504 values less than (11003906244),
partition o_ordk2505 values less than (11008300776),
partition o_ordk2506 values less than (11012695307),
partition o_ordk2507 values less than (11017089838),
partition o_ordk2508 values less than (11021484369),
partition o_ordk2509 values less than (11025878901),
partition o_ordk2510 values less than (11030273432),
partition o_ordk2511 values less than (11034667963),
partition o_ordk2512 values less than (11039062494),
partition o_ordk2513 values less than (11043457026),
partition o_ordk2514 values less than (11047851557),
partition o_ordk2515 values less than (11052246088),
partition o_ordk2516 values less than (11056640619),
partition o_ordk2517 values less than (11061035151),
partition o_ordk2518 values less than (11065429682),
partition o_ordk2519 values less than (11069824213),
partition o_ordk2520 values less than (11074218744),
partition o_ordk2521 values less than (11078613276),
partition o_ordk2522 values less than (11083007807),
partition o_ordk2523 values less than (11087402338),
partition o_ordk2524 values less than (11091796869),
partition o_ordk2525 values less than (11096191401),
partition o_ordk2526 values less than (11100585932),
partition o_ordk2527 values less than (11104980463),
partition o_ordk2528 values less than (11109374994),
partition o_ordk2529 values less than (11113769526),
partition o_ordk2530 values less than (11118164057),
partition o_ordk2531 values less than (11122558588),
partition o_ordk2532 values less than (11126953119),
partition o_ordk2533 values less than (11131347651),
partition o_ordk2534 values less than (11135742182),
partition o_ordk2535 values less than (11140136713),
partition o_ordk2536 values less than (11144531244),
partition o_ordk2537 values less than (11148925776),
partition o_ordk2538 values less than (11153320307),
partition o_ordk2539 values less than (11157714838),
partition o_ordk2540 values less than (11162109369),
partition o_ordk2541 values less than (11166503901),
partition o_ordk2542 values less than (11170898432),
partition o_ordk2543 values less than (11175292963),
partition o_ordk2544 values less than (11179687494),
partition o_ordk2545 values less than (11184082026),
partition o_ordk2546 values less than (11188476557),
partition o_ordk2547 values less than (11192871088),
partition o_ordk2548 values less than (11197265619),
partition o_ordk2549 values less than (11201660151),
partition o_ordk2550 values less than (11206054682),
partition o_ordk2551 values less than (11210449213),
partition o_ordk2552 values less than (11214843744),
partition o_ordk2553 values less than (11219238276),
partition o_ordk2554 values less than (11223632807),
partition o_ordk2555 values less than (11228027338),
partition o_ordk2556 values less than (11232421869),
partition o_ordk2557 values less than (11236816401),
partition o_ordk2558 values less than (11241210932),
partition o_ordk2559 values less than (11245605463),
partition o_ordk2560 values less than (11249999994),
partition o_ordk2561 values less than (11254394526),
partition o_ordk2562 values less than (11258789057),
partition o_ordk2563 values less than (11263183588),
partition o_ordk2564 values less than (11267578119),
partition o_ordk2565 values less than (11271972651),
partition o_ordk2566 values less than (11276367182),
partition o_ordk2567 values less than (11280761713),
partition o_ordk2568 values less than (11285156244),
partition o_ordk2569 values less than (11289550776),
partition o_ordk2570 values less than (11293945307),
partition o_ordk2571 values less than (11298339838),
partition o_ordk2572 values less than (11302734369),
partition o_ordk2573 values less than (11307128901),
partition o_ordk2574 values less than (11311523432),
partition o_ordk2575 values less than (11315917963),
partition o_ordk2576 values less than (11320312494),
partition o_ordk2577 values less than (11324707026),
partition o_ordk2578 values less than (11329101557),
partition o_ordk2579 values less than (11333496088),
partition o_ordk2580 values less than (11337890619),
partition o_ordk2581 values less than (11342285151),
partition o_ordk2582 values less than (11346679682),
```

```
partition o_ordk2583 values less than (11351074213),
partition o_ordk2584 values less than (11355468744),
partition o_ordk2585 values less than (11359863276),
partition o_ordk2586 values less than (11364257807),
partition o_ordk2587 values less than (11368652338),
partition o_ordk2588 values less than (11373046869),
partition o_ordk2589 values less than (11377441401),
partition o_ordk2590 values less than (11381835932),
partition o_ordk2591 values less than (11386230463),
partition o_ordk2592 values less than (11390624994),
partition o_ordk2593 values less than (11395019526),
partition o_ordk2594 values less than (11399414057),
partition o_ordk2595 values less than (11403808588),
partition o_ordk2596 values less than (11408203119),
partition o_ordk2597 values less than (11412597651),
partition o_ordk2598 values less than (11416992182),
partition o_ordk2599 values less than (11421386713),
partition o_ordk2600 values less than (11425781244),
partition o_ordk2601 values less than (11430175776),
partition o_ordk2602 values less than (11434570307),
partition o_ordk2603 values less than (11438964838),
partition o_ordk2604 values less than (11443359369),
partition o_ordk2605 values less than (11447753901),
partition o_ordk2606 values less than (11452148432),
partition o_ordk2607 values less than (11456542963),
partition o_ordk2608 values less than (11460937494),
partition o_ordk2609 values less than (11465332026),
partition o_ordk2610 values less than (11469726557),
partition o_ordk2611 values less than (11474121088),
partition o_ordk2612 values less than (11478515619),
partition o_ordk2613 values less than (11482910151),
partition o_ordk2614 values less than (11487304682),
partition o_ordk2615 values less than (11491699213),
partition o_ordk2616 values less than (11496093744),
partition o_ordk2617 values less than (11500488275),
partition o_ordk2618 values less than (11504882807),
partition o_ordk2619 values less than (11509277338),
partition o_ordk2620 values less than (11513671869),
partition o_ordk2621 values less than (11518066400),
partition o_ordk2622 values less than (11522460932),
partition o_ordk2623 values less than (11526855463),
partition o_ordk2624 values less than (11531249994),
partition o_ordk2625 values less than (11535644525),
partition o_ordk2626 values less than (11540039057),
partition o_ordk2627 values less than (11544433588),
partition o_ordk2628 values less than (11548828119),
partition o_ordk2629 values less than (11553222650),
partition o_ordk2630 values less than (11557617182),
partition o_ordk2631 values less than (11562011713),
partition o_ordk2632 values less than (11566406244),
partition o_ordk2633 values less than (11570800775),
partition o_ordk2634 values less than (11575195307),
partition o_ordk2635 values less than (11579589838),
partition o_ordk2636 values less than (11583984369),
partition o_ordk2637 values less than (11588378900),
partition o_ordk2638 values less than (11592773432),
partition o_ordk2639 values less than (11597167963),
partition o_ordk2640 values less than (11601562494),
partition o_ordk2641 values less than (11605957025),
partition o_ordk2642 values less than (11610351557),
partition o_ordk2643 values less than (11614746088),
partition o_ordk2644 values less than (11619140619),
partition o_ordk2645 values less than (11623535150),
partition o_ordk2646 values less than (11627929682),
partition o_ordk2647 values less than (11632324213),
partition o_ordk2648 values less than (11636718744),
partition o_ordk2649 values less than (11641113275),
partition o_ordk2650 values less than (11645507807),
partition o_ordk2651 values less than (11649902338),
partition o_ordk2652 values less than (11654296869),
partition o_ordk2653 values less than (11658691400),
partition o_ordk2654 values less than (11663085932),
partition o_ordk2655 values less than (11667480463),
partition o_ordk2656 values less than (11671874994),
partition o_ordk2657 values less than (11676269525),
partition o_ordk2658 values less than (11680664057),
partition o_ordk2659 values less than (11685058588),
partition o_ordk2660 values less than (11689453119),
partition o_ordk2661 values less than (11693847650),
partition o_ordk2662 values less than (11698242182),
partition o_ordk2663 values less than (11702636713),
partition o_ordk2664 values less than (11707031244),
partition o_ordk2665 values less than (11711425775),
partition o_ordk2666 values less than (11715820307),
partition o_ordk2667 values less than (11720214838),
partition o_ordk2668 values less than (11724609369),
partition o_ordk2669 values less than (11729003900),
partition o_ordk2670 values less than (11733398432),
partition o_ordk2671 values less than (11737792963),
partition o_ordk2672 values less than (11742187494),
partition o_ordk2673 values less than (11746582025),
partition o_ordk2674 values less than (11750976557),
partition o_ordk2675 values less than (11755371088),
partition o_ordk2676 values less than (11759765619),
partition o_ordk2677 values less than (11764160150),
partition o_ordk2678 values less than (11768554682),
partition o_ordk2679 values less than (11772949213),
partition o_ordk2680 values less than (11777343744),
partition o_ordk2681 values less than (11781738275),
partition o_ordk2682 values less than (11786132807),
partition o_ordk2683 values less than (11790527338),
partition o_ordk2684 values less than (11794921869),
partition o_ordk2685 values less than (11799316400),
partition o_ordk2686 values less than (11803710932),
partition o_ordk2687 values less than (11808105463),
partition o_ordk2688 values less than (11812499994),
partition o_ordk2689 values less than (11816894525),
partition o_ordk2690 values less than (11821289057),
partition o_ordk2691 values less than (11825683588),
partition o_ordk2692 values less than (11830078119),
partition o_ordk2693 values less than (11834472650),
partition o_ordk2694 values less than (11838867182),
partition o_ordk2695 values less than (11843261713),
partition o_ordk2696 values less than (11847656244),
partition o_ordk2697 values less than (11852050775),
partition o_ordk2698 values less than (11856445307),
partition o_ordk2699 values less than (11860839838),
partition o_ordk2700 values less than (11865234369),
partition o_ordk2701 values less than (11869628900),
partition o_ordk2702 values less than (11874023432),
partition o_ordk2703 values less than (11878417963),
partition o_ordk2704 values less than (11882812494),
partition o_ordk2705 values less than (11887207025),
partition o_ordk2706 values less than (11891601557),
partition o_ordk2707 values less than (11895996088),
partition o_ordk2708 values less than (11900390619),
partition o_ordk2709 values less than (11904785150),
partition o_ordk2710 values less than (11909179682),
partition o_ordk2711 values less than (11913574213),
partition o_ordk2712 values less than (11917968744),
partition o_ordk2713 values less than (11922363275),
partition o_ordk2714 values less than (11926757807),
partition o_ordk2715 values less than (11931152338),
partition o_ordk2716 values less than (11935546869),
partition o_ordk2717 values less than (11939941400),
partition o_ordk2718 values less than (11944335932),
partition o_ordk2719 values less than (11948730463),
partition o_ordk2720 values less than (11953124994),
partition o_ordk2721 values less than (11957519525),
partition o_ordk2722 values less than (11961914057),
partition o_ordk2723 values less than (11966308588),
partition o_ordk2724 values less than (11970703119),
partition o_ordk2725 values less than (11975097650),
partition o_ordk2726 values less than (11979492182),
partition o_ordk2727 values less than (11983886713),
partition o_ordk2728 values less than (11988281244),
partition o_ordk2729 values less than (11992675775),
partition o_ordk2730 values less than (11997070307),
partition o_ordk2731 values less than (12001464838),
partition o_ordk2732 values less than (12005859369),
partition o_ordk2733 values less than (12010253900),
partition o_ordk2734 values less than (12014648431),
partition o_ordk2735 values less than (12019042963),
partition o_ordk2736 values less than (12023437494),
partition o_ordk2737 values less than (12027832025),
partition o_ordk2738 values less than (12032226556),
partition o_ordk2739 values less than (12036621088),
partition o_ordk2740 values less than (12041015619),
partition o_ordk2741 values less than (12045410150),
partition o_ordk2742 values less than (12049804681),
partition o_ordk2743 values less than (12054199213),
partition o_ordk2744 values less than (12058593744),
partition o_ordk2745 values less than (12062988275),
partition o_ordk2746 values less than (12067382806),
partition o_ordk2747 values less than (12071777338),
partition o_ordk2748 values less than (12076171869),
partition o_ordk2749 values less than (12080566400),
partition o_ordk2750 values less than (12084960931),
partition o_ordk2751 values less than (12089355463),
partition o_ordk2752 values less than (12093749994),
partition o_ordk2753 values less than (12098144525),
partition o_ordk2754 values less than (12102539056),
partition o_ordk2755 values less than (12106933588),
partition o_ordk2756 values less than (12111328119),
partition o_ordk2757 values less than (12115722650),
partition o_ordk2758 values less than (12120117181),
partition o_ordk2759 values less than (12124511713),
partition o_ordk2760 values less than (12128906244),
partition o_ordk2761 values less than (12133300775),
partition o_ordk2762 values less than (12137695306),
```

```
partition o_ordk2763 values less than (12142089838),
partition o_ordk2764 values less than (12146484369),
partition o_ordk2765 values less than (12150878900),
partition o_ordk2766 values less than (12155273431),
partition o_ordk2767 values less than (12159667963),
partition o_ordk2768 values less than (12164062494),
partition o_ordk2769 values less than (12168457025),
partition o_ordk2770 values less than (12172851556),
partition o_ordk2771 values less than (12177246088),
partition o_ordk2772 values less than (12181640619),
partition o_ordk2773 values less than (12186035150),
partition o_ordk2774 values less than (12190429681),
partition o_ordk2775 values less than (12194824213),
partition o_ordk2776 values less than (12199218744),
partition o_ordk2777 values less than (12203613275),
partition o_ordk2778 values less than (12208007806),
partition o_ordk2779 values less than (12212402338),
partition o_ordk2780 values less than (12216796869),
partition o_ordk2781 values less than (12221191400),
partition o_ordk2782 values less than (12225585931),
partition o_ordk2783 values less than (12229980463),
partition o_ordk2784 values less than (12234374994),
partition o_ordk2785 values less than (12238769525),
partition o_ordk2786 values less than (12243164056),
partition o_ordk2787 values less than (12247558588),
partition o_ordk2788 values less than (12251953119),
partition o_ordk2789 values less than (12256347650),
partition o_ordk2790 values less than (12260742181),
partition o_ordk2791 values less than (12265136713),
partition o_ordk2792 values less than (12269531244),
partition o_ordk2793 values less than (12273925775),
partition o_ordk2794 values less than (12278320306),
partition o_ordk2795 values less than (12282714838),
partition o_ordk2796 values less than (12287109369),
partition o_ordk2797 values less than (12291503900),
partition o_ordk2798 values less than (12295898431),
partition o_ordk2799 values less than (12300292963),
partition o_ordk2800 values less than (12304687494),
partition o_ordk2801 values less than (12309082025),
partition o_ordk2802 values less than (12313476556),
partition o_ordk2803 values less than (12317871088),
partition o_ordk2804 values less than (12322265619),
partition o_ordk2805 values less than (12326660150),
partition o_ordk2806 values less than (12331054681),
partition o_ordk2807 values less than (12335449213),
partition o_ordk2808 values less than (12339843744),
partition o_ordk2809 values less than (12344238275),
partition o_ordk2810 values less than (12348632806),
partition o_ordk2811 values less than (12353027338),
partition o_ordk2812 values less than (12357421869),
partition o_ordk2813 values less than (12361816400),
partition o_ordk2814 values less than (12366210931),
partition o_ordk2815 values less than (12370605463),
partition o_ordk2816 values less than (12374999994),
partition o_ordk2817 values less than (12379394525),
partition o_ordk2818 values less than (12383789056),
partition o_ordk2819 values less than (12388183588),
partition o_ordk2820 values less than (12392578119),
partition o_ordk2821 values less than (12396972650),
partition o_ordk2822 values less than (12401367181),
partition o_ordk2823 values less than (12405761713),
partition o_ordk2824 values less than (12410156244),
partition o_ordk2825 values less than (12414550775),
partition o_ordk2826 values less than (12418945306),
partition o_ordk2827 values less than (12423339838),
partition o_ordk2828 values less than (12427734369),
partition o_ordk2829 values less than (12432128900),
partition o_ordk2830 values less than (12436523431),
partition o_ordk2831 values less than (12440917963),
partition o_ordk2832 values less than (12445312494),
partition o_ordk2833 values less than (12449707025),
partition o_ordk2834 values less than (12454101556),
partition o_ordk2835 values less than (12458496088),
partition o_ordk2836 values less than (12462890619),
partition o_ordk2837 values less than (12467285150),
partition o_ordk2838 values less than (12471679681),
partition o_ordk2839 values less than (12476074213),
partition o_ordk2840 values less than (12480468744),
partition o_ordk2841 values less than (12484863275),
partition o_ordk2842 values less than (12489257806),
partition o_ordk2843 values less than (12493652338),
partition o_ordk2844 values less than (12498046869),
partition o_ordk2845 values less than (12502441400),
partition o_ordk2846 values less than (12506835931),
partition o_ordk2847 values less than (12511230462),
partition o_ordk2848 values less than (12515624994),
partition o_ordk2849 values less than (12520019525),
partition o_ordk2850 values less than (12524414056),
partition o_ordk2851 values less than (12528808587),
partition o_ordk2852 values less than (12533203119),
partition o_ordk2853 values less than (12537597650),
partition o_ordk2854 values less than (12541992181),
partition o_ordk2855 values less than (12546386712),
partition o_ordk2856 values less than (12550781244),
partition o_ordk2857 values less than (12555175775),
partition o_ordk2858 values less than (12559570306),
partition o_ordk2859 values less than (12563964837),
partition o_ordk2860 values less than (12568359369),
partition o_ordk2861 values less than (12572753900),
partition o_ordk2862 values less than (12577148431),
partition o_ordk2863 values less than (12581542962),
partition o_ordk2864 values less than (12585937494),
partition o_ordk2865 values less than (12590332025),
partition o_ordk2866 values less than (12594726556),
partition o_ordk2867 values less than (12599121087),
partition o_ordk2868 values less than (12603515619),
partition o_ordk2869 values less than (12607910150),
partition o_ordk2870 values less than (12612304681),
partition o_ordk2871 values less than (12616699212),
partition o_ordk2872 values less than (12621093744),
partition o_ordk2873 values less than (12625488275),
partition o_ordk2874 values less than (12629882806),
partition o_ordk2875 values less than (12634277337),
partition o_ordk2876 values less than (12638671869),
partition o_ordk2877 values less than (12643066400),
partition o_ordk2878 values less than (12647460931),
partition o_ordk2879 values less than (12651855462),
partition o_ordk2880 values less than (12656249994),
partition o_ordk2881 values less than (12660644525),
partition o_ordk2882 values less than (12665039056),
partition o_ordk2883 values less than (12669433587),
partition o_ordk2884 values less than (12673828119),
partition o_ordk2885 values less than (12678222650),
partition o_ordk2886 values less than (12682617181),
partition o_ordk2887 values less than (12687011712),
partition o_ordk2888 values less than (12691406244),
partition o_ordk2889 values less than (12695800775),
partition o_ordk2890 values less than (12700195306),
partition o_ordk2891 values less than (12704589837),
partition o_ordk2892 values less than (12708984369),
partition o_ordk2893 values less than (12713378900),
partition o_ordk2894 values less than (12717773431),
partition o_ordk2895 values less than (12722167962),
partition o_ordk2896 values less than (12726562494),
partition o_ordk2897 values less than (12730957025),
partition o_ordk2898 values less than (12735351556),
partition o_ordk2899 values less than (12739746087),
partition o_ordk2900 values less than (12744140619),
partition o_ordk2901 values less than (12748535150),
partition o_ordk2902 values less than (12752929681),
partition o_ordk2903 values less than (12757324212),
partition o_ordk2904 values less than (12761718744),
partition o_ordk2905 values less than (12766113275),
partition o_ordk2906 values less than (12770507806),
partition o_ordk2907 values less than (12774902337),
partition o_ordk2908 values less than (12779296869),
partition o_ordk2909 values less than (12783691400),
partition o_ordk2910 values less than (12788085931),
partition o_ordk2911 values less than (12792480462),
partition o_ordk2912 values less than (12796874994),
partition o_ordk2913 values less than (12801269525),
partition o_ordk2914 values less than (12805664056),
partition o_ordk2915 values less than (12810058587),
partition o_ordk2916 values less than (12814453119),
partition o_ordk2917 values less than (12818847650),
partition o_ordk2918 values less than (12823242181),
partition o_ordk2919 values less than (12827636712),
partition o_ordk2920 values less than (12832031244),
partition o_ordk2921 values less than (12836425775),
partition o_ordk2922 values less than (12840820306),
partition o_ordk2923 values less than (12845214837),
partition o_ordk2924 values less than (12849609369),
partition o_ordk2925 values less than (12854003900),
partition o_ordk2926 values less than (12858398431),
partition o_ordk2927 values less than (12862792962),
partition o_ordk2928 values less than (12867187494),
partition o_ordk2929 values less than (12871582025),
partition o_ordk2930 values less than (12875976556),
partition o_ordk2931 values less than (12880371087),
partition o_ordk2932 values less than (12884765619),
partition o_ordk2933 values less than (12889160150),
partition o_ordk2934 values less than (12893554681),
partition o_ordk2935 values less than (12897949212),
partition o_ordk2936 values less than (12902343744),
partition o_ordk2937 values less than (12906738275),
partition o_ordk2938 values less than (12911132806),
partition o_ordk2939 values less than (12915527337),
partition o_ordk2940 values less than (12919921869),
partition o_ordk2941 values less than (12924316400),
partition o_ordk2942 values less than (12928710931),
```

```
partition o_ordk2943 values less than (12933105462),
partition o_ordk2944 values less than (12937499994),
partition o_ordk2945 values less than (12941894525),
partition o_ordk2946 values less than (12946289056),
partition o_ordk2947 values less than (12950683587),
partition o_ordk2948 values less than (12955078119),
partition o_ordk2949 values less than (12959472650),
partition o_ordk2950 values less than (12963867181),
partition o_ordk2951 values less than (12968261712),
partition o_ordk2952 values less than (12972656244),
partition o_ordk2953 values less than (12977050775),
partition o_ordk2954 values less than (12981445306),
partition o_ordk2955 values less than (12985839837),
partition o_ordk2956 values less than (12990234369),
partition o_ordk2957 values less than (12994628900),
partition o_ordk2958 values less than (12999023431),
partition o_ordk2959 values less than (13003417962),
partition o_ordk2960 values less than (13007812493),
partition o_ordk2961 values less than (13012207025),
partition o_ordk2962 values less than (13016601556),
partition o_ordk2963 values less than (13020996087),
partition o_ordk2964 values less than (13025390618),
partition o_ordk2965 values less than (13029785150),
partition o_ordk2966 values less than (13034179681),
partition o_ordk2967 values less than (13038574212),
partition o_ordk2968 values less than (13042968743),
partition o_ordk2969 values less than (13047363275),
partition o_ordk2970 values less than (13051757806),
partition o_ordk2971 values less than (13056152337),
partition o_ordk2972 values less than (13060546868),
partition o_ordk2973 values less than (13064941400),
partition o_ordk2974 values less than (13069335931),
partition o_ordk2975 values less than (13073730462),
partition o_ordk2976 values less than (13078124993),
partition o_ordk2977 values less than (13082519525),
partition o_ordk2978 values less than (13086914056),
partition o_ordk2979 values less than (13091308587),
partition o_ordk2980 values less than (13095703118),
partition o_ordk2981 values less than (13100097650),
partition o_ordk2982 values less than (13104492181),
partition o_ordk2983 values less than (13108886712),
partition o_ordk2984 values less than (13113281243),
partition o_ordk2985 values less than (13117675775),
partition o_ordk2986 values less than (13122070306),
partition o_ordk2987 values less than (13126464837),
partition o_ordk2988 values less than (13130859368),
partition o_ordk2989 values less than (13135253900),
partition o_ordk2990 values less than (13139648431),
partition o_ordk2991 values less than (13144042962),
partition o_ordk2992 values less than (13148437493),
partition o_ordk2993 values less than (13152832025),
partition o_ordk2994 values less than (13157226556),
partition o_ordk2995 values less than (13161621087),
partition o_ordk2996 values less than (13166015618),
partition o_ordk2997 values less than (13170410150),
partition o_ordk2998 values less than (13174804681),
partition o_ordk2999 values less than (13179199212),
partition o_ordk3000 values less than (13183593743),
partition o_ordk3001 values less than (13187988275),
partition o_ordk3002 values less than (13192382806),
partition o_ordk3003 values less than (13196777337),
partition o_ordk3004 values less than (13201171868),
partition o_ordk3005 values less than (13205566400),
partition o_ordk3006 values less than (13209960931),
partition o_ordk3007 values less than (13214355462),
partition o_ordk3008 values less than (13218749993),
partition o_ordk3009 values less than (13223144525),
partition o_ordk3010 values less than (13227539056),
partition o_ordk3011 values less than (13231933587),
partition o_ordk3012 values less than (13236328118),
partition o_ordk3013 values less than (13240722650),
partition o_ordk3014 values less than (13245117181),
partition o_ordk3015 values less than (13249511712),
partition o_ordk3016 values less than (13253906243),
partition o_ordk3017 values less than (13258300775),
partition o_ordk3018 values less than (13262695306),
partition o_ordk3019 values less than (13267089837),
partition o_ordk3020 values less than (13271484368),
partition o_ordk3021 values less than (13275878900),
partition o_ordk3022 values less than (13280273431),
partition o_ordk3023 values less than (13284667962),
partition o_ordk3024 values less than (13289062493),
partition o_ordk3025 values less than (13293457025),
partition o_ordk3026 values less than (13297851556),
partition o_ordk3027 values less than (13302246087),
partition o_ordk3028 values less than (13306640618),
partition o_ordk3029 values less than (13311035150),
partition o_ordk3030 values less than (13315429681),
partition o_ordk3031 values less than (13319824212),
partition o_ordk3032 values less than (13324218743),
partition o_ordk3033 values less than (13328613275),
partition o_ordk3034 values less than (13333007806),
partition o_ordk3035 values less than (13337402337),
partition o_ordk3036 values less than (13341796868),
partition o_ordk3037 values less than (13346191400),
partition o_ordk3038 values less than (13350585931),
partition o_ordk3039 values less than (13354980462),
partition o_ordk3040 values less than (13359374993),
partition o_ordk3041 values less than (13363769525),
partition o_ordk3042 values less than (13368164056),
partition o_ordk3043 values less than (13372558587),
partition o_ordk3044 values less than (13376953118),
partition o_ordk3045 values less than (13381347650),
partition o_ordk3046 values less than (13385742181),
partition o_ordk3047 values less than (13390136712),
partition o_ordk3048 values less than (13394531243),
partition o_ordk3049 values less than (13398925775),
partition o_ordk3050 values less than (13403320306),
partition o_ordk3051 values less than (13407714837),
partition o_ordk3052 values less than (13412109368),
partition o_ordk3053 values less than (13416503900),
partition o_ordk3054 values less than (13420898431),
partition o_ordk3055 values less than (13425292962),
partition o_ordk3056 values less than (13429687493),
partition o_ordk3057 values less than (13434082025),
partition o_ordk3058 values less than (13438476556),
partition o_ordk3059 values less than (13442871087),
partition o_ordk3060 values less than (13447265618),
partition o_ordk3061 values less than (13451660150),
partition o_ordk3062 values less than (13456054681),
partition o_ordk3063 values less than (13460449212),
partition o_ordk3064 values less than (13464843743),
partition o_ordk3065 values less than (13469238275),
partition o_ordk3066 values less than (13473632806),
partition o_ordk3067 values less than (13478027337),
partition o_ordk3068 values less than (13482421868),
partition o_ordk3069 values less than (13486816400),
partition o_ordk3070 values less than (13491210931),
partition o_ordk3071 values less than (13495605462),
partition o_ordk3072 values less than (13499999993),
partition o_ordk3073 values less than (13504394524),
partition o_ordk3074 values less than (13508789056),
partition o_ordk3075 values less than (13513183587),
partition o_ordk3076 values less than (13517578118),
partition o_ordk3077 values less than (13521972649),
partition o_ordk3078 values less than (13526367181),
partition o_ordk3079 values less than (13530761712),
partition o_ordk3080 values less than (13535156243),
partition o_ordk3081 values less than (13539550774),
partition o_ordk3082 values less than (13543945306),
partition o_ordk3083 values less than (13548339837),
partition o_ordk3084 values less than (13552734368),
partition o_ordk3085 values less than (13557128899),
partition o_ordk3086 values less than (13561523431),
partition o_ordk3087 values less than (13565917962),
partition o_ordk3088 values less than (13570312493),
partition o_ordk3089 values less than (13574707024),
partition o_ordk3090 values less than (13579101556),
partition o_ordk3091 values less than (13583496087),
partition o_ordk3092 values less than (13587890618),
partition o_ordk3093 values less than (13592285149),
partition o_ordk3094 values less than (13596679681),
partition o_ordk3095 values less than (13601074212),
partition o_ordk3096 values less than (13605468743),
partition o_ordk3097 values less than (13609863274),
partition o_ordk3098 values less than (13614257806),
partition o_ordk3099 values less than (13618652337),
partition o_ordk3100 values less than (13623046868),
partition o_ordk3101 values less than (13627441399),
partition o_ordk3102 values less than (13631835931),
partition o_ordk3103 values less than (13636230462),
partition o_ordk3104 values less than (13640624993),
partition o_ordk3105 values less than (13645019524),
partition o_ordk3106 values less than (13649414056),
partition o_ordk3107 values less than (13653808587),
partition o_ordk3108 values less than (13658203118),
partition o_ordk3109 values less than (13662597649),
partition o_ordk3110 values less than (13666992181),
partition o_ordk3111 values less than (13671386712),
partition o_ordk3112 values less than (13675781243),
partition o_ordk3113 values less than (13680175774),
partition o_ordk3114 values less than (13684570306),
partition o_ordk3115 values less than (13688964837),
partition o_ordk3116 values less than (13693359368),
partition o_ordk3117 values less than (13697753899),
partition o_ordk3118 values less than (13702148431),
partition o_ordk3119 values less than (13706542962),
partition o_ordk3120 values less than (13710937493),
partition o_ordk3121 values less than (13715332024),
partition o_ordk3122 values less than (13719726556),
```

```
partition o_ordk3123 values less than (13724121087),
partition o_ordk3124 values less than (13728515618),
partition o_ordk3125 values less than (13732910149),
partition o_ordk3126 values less than (13737304681),
partition o_ordk3127 values less than (13741699212),
partition o_ordk3128 values less than (13746093743),
partition o_ordk3129 values less than (13750488274),
partition o_ordk3130 values less than (13754882806),
partition o_ordk3131 values less than (13759277337),
partition o_ordk3132 values less than (13763671868),
partition o_ordk3133 values less than (13768066399),
partition o_ordk3134 values less than (13772460931),
partition o_ordk3135 values less than (13776855462),
partition o_ordk3136 values less than (13781249993),
partition o_ordk3137 values less than (13785644524),
partition o_ordk3138 values less than (13790039056),
partition o_ordk3139 values less than (13794433587),
partition o_ordk3140 values less than (13798828118),
partition o_ordk3141 values less than (13803222649),
partition o_ordk3142 values less than (13807617181),
partition o_ordk3143 values less than (13812011712),
partition o_ordk3144 values less than (13816406243),
partition o_ordk3145 values less than (13820800774),
partition o_ordk3146 values less than (13825195306),
partition o_ordk3147 values less than (13829589837),
partition o_ordk3148 values less than (13833984368),
partition o_ordk3149 values less than (13838378899),
partition o_ordk3150 values less than (13842773431),
partition o_ordk3151 values less than (13847167962),
partition o_ordk3152 values less than (13851562493),
partition o_ordk3153 values less than (13855957024),
partition o_ordk3154 values less than (13860351556),
partition o_ordk3155 values less than (13864746087),
partition o_ordk3156 values less than (13869140618),
partition o_ordk3157 values less than (13873535149),
partition o_ordk3158 values less than (13877929681),
partition o_ordk3159 values less than (13882324212),
partition o_ordk3160 values less than (13886718743),
partition o_ordk3161 values less than (13891113274),
partition o_ordk3162 values less than (13895507806),
partition o_ordk3163 values less than (13899902337),
partition o_ordk3164 values less than (13904296868),
partition o_ordk3165 values less than (13908691399),
partition o_ordk3166 values less than (13913085931),
partition o_ordk3167 values less than (13917480462),
partition o_ordk3168 values less than (13921874993),
partition o_ordk3169 values less than (13926269524),
partition o_ordk3170 values less than (13930664056),
partition o_ordk3171 values less than (13935058587),
partition o_ordk3172 values less than (13939453118),
partition o_ordk3173 values less than (13943847649),
partition o_ordk3174 values less than (13948242181),
partition o_ordk3175 values less than (13952636712),
partition o_ordk3176 values less than (13957031243),
partition o_ordk3177 values less than (13961425774),
partition o_ordk3178 values less than (13965820306),
partition o_ordk3179 values less than (13970214837),
partition o_ordk3180 values less than (13974609368),
partition o_ordk3181 values less than (13979003899),
partition o_ordk3182 values less than (13983398431),
partition o_ordk3183 values less than (13987792962),
partition o_ordk3184 values less than (13992187493),
partition o_ordk3185 values less than (13996582024),
partition o_ordk3186 values less than (14000976555),
partition o_ordk3187 values less than (14005371087),
partition o_ordk3188 values less than (14009765618),
partition o_ordk3189 values less than (14014160149),
partition o_ordk3190 values less than (14018554680),
partition o_ordk3191 values less than (14022949212),
partition o_ordk3192 values less than (14027343743),
partition o_ordk3193 values less than (14031738274),
partition o_ordk3194 values less than (14036132805),
partition o_ordk3195 values less than (14040527337),
partition o_ordk3196 values less than (14044921868),
partition o_ordk3197 values less than (14049316399),
partition o_ordk3198 values less than (14053710930),
partition o_ordk3199 values less than (14058105462),
partition o_ordk3200 values less than (14062499993),
partition o_ordk3201 values less than (14066894524),
partition o_ordk3202 values less than (14071289055),
partition o_ordk3203 values less than (14075683587),
partition o_ordk3204 values less than (14080078118),
partition o_ordk3205 values less than (14084472649),
partition o_ordk3206 values less than (14088867180),
partition o_ordk3207 values less than (14093261712),
partition o_ordk3208 values less than (14097656243),
partition o_ordk3209 values less than (14102050774),
partition o_ordk3210 values less than (14106445305),
partition o_ordk3211 values less than (14110839837),
partition o_ordk3212 values less than (14115234368),
partition o_ordk3213 values less than (14119628899),
partition o_ordk3214 values less than (14124023430),
partition o_ordk3215 values less than (14128417962),
partition o_ordk3216 values less than (14132812493),
partition o_ordk3217 values less than (14137207024),
partition o_ordk3218 values less than (14141601555),
partition o_ordk3219 values less than (14145996087),
partition o_ordk3220 values less than (14150390618),
partition o_ordk3221 values less than (14154785149),
partition o_ordk3222 values less than (14159179680),
partition o_ordk3223 values less than (14163574212),
partition o_ordk3224 values less than (14167968743),
partition o_ordk3225 values less than (14172363274),
partition o_ordk3226 values less than (14176757805),
partition o_ordk3227 values less than (14181152337),
partition o_ordk3228 values less than (14185546868),
partition o_ordk3229 values less than (14189941399),
partition o_ordk3230 values less than (14194335930),
partition o_ordk3231 values less than (14198730462),
partition o_ordk3232 values less than (14203124993),
partition o_ordk3233 values less than (14207519524),
partition o_ordk3234 values less than (14211914055),
partition o_ordk3235 values less than (14216308587),
partition o_ordk3236 values less than (14220703118),
partition o_ordk3237 values less than (14225097649),
partition o_ordk3238 values less than (14229492180),
partition o_ordk3239 values less than (14233886712),
partition o_ordk3240 values less than (14238281243),
partition o_ordk3241 values less than (14242675774),
partition o_ordk3242 values less than (14247070305),
partition o_ordk3243 values less than (14251464837),
partition o_ordk3244 values less than (14255859368),
partition o_ordk3245 values less than (14260253899),
partition o_ordk3246 values less than (14264648430),
partition o_ordk3247 values less than (14269042962),
partition o_ordk3248 values less than (14273437493),
partition o_ordk3249 values less than (14277832024),
partition o_ordk3250 values less than (14282226555),
partition o_ordk3251 values less than (14286621087),
partition o_ordk3252 values less than (14291015618),
partition o_ordk3253 values less than (14295410149),
partition o_ordk3254 values less than (14299804680),
partition o_ordk3255 values less than (14304199212),
partition o_ordk3256 values less than (14308593743),
partition o_ordk3257 values less than (14312988274),
partition o_ordk3258 values less than (14317382805),
partition o_ordk3259 values less than (14321777337),
partition o_ordk3260 values less than (14326171868),
partition o_ordk3261 values less than (14330566399),
partition o_ordk3262 values less than (14334960930),
partition o_ordk3263 values less than (14339355462),
partition o_ordk3264 values less than (14343749993),
partition o_ordk3265 values less than (14348144524),
partition o_ordk3266 values less than (14352539055),
partition o_ordk3267 values less than (14356933587),
partition o_ordk3268 values less than (14361328118),
partition o_ordk3269 values less than (14365722649),
partition o_ordk3270 values less than (14370117180),
partition o_ordk3271 values less than (14374511712),
partition o_ordk3272 values less than (14378906243),
partition o_ordk3273 values less than (14383300774),
partition o_ordk3274 values less than (14387695305),
partition o_ordk3275 values less than (14392089837),
partition o_ordk3276 values less than (14396484368),
partition o_ordk3277 values less than (14400878899),
partition o_ordk3278 values less than (14405273430),
partition o_ordk3279 values less than (14409667962),
partition o_ordk3280 values less than (14414062493),
partition o_ordk3281 values less than (14418457024),
partition o_ordk3282 values less than (14422851555),
partition o_ordk3283 values less than (14427246087),
partition o_ordk3284 values less than (14431640618),
partition o_ordk3285 values less than (14436035149),
partition o_ordk3286 values less than (14440429680),
partition o_ordk3287 values less than (14444824212),
partition o_ordk3288 values less than (14449218743),
partition o_ordk3289 values less than (14453613274),
partition o_ordk3290 values less than (14458007805),
partition o_ordk3291 values less than (14462402337),
partition o_ordk3292 values less than (14466796868),
partition o_ordk3293 values less than (14471191399),
partition o_ordk3294 values less than (14475585930),
partition o_ordk3295 values less than (14479980462),
partition o_ordk3296 values less than (14484374993),
partition o_ordk3297 values less than (14488769524),
partition o_ordk3298 values less than (14493164055),
partition o_ordk3299 values less than (14497558587),
partition o_ordk3300 values less than (14501953118),
partition o_ordk3301 values less than (14506347649),
partition o_ordk3302 values less than (14510742180),
```

```
partition o_ordk3303 values less than (14515136711),
partition o_ordk3304 values less than (14519531243),
partition o_ordk3305 values less than (14523925774),
partition o_ordk3306 values less than (14528320305),
partition o_ordk3307 values less than (14532714836),
partition o_ordk3308 values less than (14537109368),
partition o_ordk3309 values less than (14541503899),
partition o_ordk3310 values less than (14545898430),
partition o_ordk3311 values less than (14550292961),
partition o_ordk3312 values less than (14554687493),
partition o_ordk3313 values less than (14559082024),
partition o_ordk3314 values less than (14563476555),
partition o_ordk3315 values less than (14567871086),
partition o_ordk3316 values less than (14572265618),
partition o_ordk3317 values less than (14576660149),
partition o_ordk3318 values less than (14581054680),
partition o_ordk3319 values less than (14585449211),
partition o_ordk3320 values less than (14589843743),
partition o_ordk3321 values less than (14594238274),
partition o_ordk3322 values less than (14598632805),
partition o_ordk3323 values less than (14603027336),
partition o_ordk3324 values less than (14607421868),
partition o_ordk3325 values less than (14611816399),
partition o_ordk3326 values less than (14616210930),
partition o_ordk3327 values less than (14620605461),
partition o_ordk3328 values less than (14624999993),
partition o_ordk3329 values less than (14629394524),
partition o_ordk3330 values less than (14633789055),
partition o_ordk3331 values less than (14638183586),
partition o_ordk3332 values less than (14642578118),
partition o_ordk3333 values less than (14646972649),
partition o_ordk3334 values less than (14651367180),
partition o_ordk3335 values less than (14655761711),
partition o_ordk3336 values less than (14660156243),
partition o_ordk3337 values less than (14664550774),
partition o_ordk3338 values less than (14668945305),
partition o_ordk3339 values less than (14673339836),
partition o_ordk3340 values less than (14677734368),
partition o_ordk3341 values less than (14682128899),
partition o_ordk3342 values less than (14686523430),
partition o_ordk3343 values less than (14690917961),
partition o_ordk3344 values less than (14695312493),
partition o_ordk3345 values less than (14699707024),
partition o_ordk3346 values less than (14704101555),
partition o_ordk3347 values less than (14708496086),
partition o_ordk3348 values less than (14712890618),
partition o_ordk3349 values less than (14717285149),
partition o_ordk3350 values less than (14721679680),
partition o_ordk3351 values less than (14726074211),
partition o_ordk3352 values less than (14730468743),
partition o_ordk3353 values less than (14734863274),
partition o_ordk3354 values less than (14739257805),
partition o_ordk3355 values less than (14743652336),
partition o_ordk3356 values less than (14748046868),
partition o_ordk3357 values less than (14752441399),
partition o_ordk3358 values less than (14756835930),
partition o_ordk3359 values less than (14761230461),
partition o_ordk3360 values less than (14765624993),
partition o_ordk3361 values less than (14770019524),
partition o_ordk3362 values less than (14774414055),
partition o_ordk3363 values less than (14778808586),
partition o_ordk3364 values less than (14783203118),
partition o_ordk3365 values less than (14787597649),
partition o_ordk3366 values less than (14791992180),
partition o_ordk3367 values less than (14796386711),
partition o_ordk3368 values less than (14800781243),
partition o_ordk3369 values less than (14805175774),
partition o_ordk3370 values less than (14809570305),
partition o_ordk3371 values less than (14813964836),
partition o_ordk3372 values less than (14818359368),
partition o_ordk3373 values less than (14822753899),
partition o_ordk3374 values less than (14827148430),
partition o_ordk3375 values less than (14831542961),
partition o_ordk3376 values less than (14835937493),
partition o_ordk3377 values less than (14840332024),
partition o_ordk3378 values less than (14844726555),
partition o_ordk3379 values less than (14849121086),
partition o_ordk3380 values less than (14853515618),
partition o_ordk3381 values less than (14857910149),
partition o_ordk3382 values less than (14862304680),
partition o_ordk3383 values less than (14866699211),
partition o_ordk3384 values less than (14871093743),
partition o_ordk3385 values less than (14875488274),
partition o_ordk3386 values less than (14879882805),
partition o_ordk3387 values less than (14884277336),
partition o_ordk3388 values less than (14888671868),
partition o_ordk3389 values less than (14893066399),
partition o_ordk3390 values less than (14897460930),
partition o_ordk3391 values less than (14901855461),
partition o_ordk3392 values less than (14906249993),
partition o_ordk3393 values less than (14910644524),
partition o_ordk3394 values less than (14915039055),
partition o_ordk3395 values less than (14919433586),
partition o_ordk3396 values less than (14923828118),
partition o_ordk3397 values less than (14928222649),
partition o_ordk3398 values less than (14932617180),
partition o_ordk3399 values less than (14937011711),
partition o_ordk3400 values less than (14941406243),
partition o_ordk3401 values less than (14945800774),
partition o_ordk3402 values less than (14950195305),
partition o_ordk3403 values less than (14954589836),
partition o_ordk3404 values less than (14958984368),
partition o_ordk3405 values less than (14963378899),
partition o_ordk3406 values less than (14967773430),
partition o_ordk3407 values less than (14972167961),
partition o_ordk3408 values less than (14976562493),
partition o_ordk3409 values less than (14980957024),
partition o_ordk3410 values less than (14985351555),
partition o_ordk3411 values less than (14989746086),
partition o_ordk3412 values less than (14994140618),
partition o_ordk3413 values less than (14998535149),
partition o_ordk3414 values less than (15002929680),
partition o_ordk3415 values less than (15007324211),
partition o_ordk3416 values less than (15011718742),
partition o_ordk3417 values less than (15016113274),
partition o_ordk3418 values less than (15020507805),
partition o_ordk3419 values less than (15024902336),
partition o_ordk3420 values less than (15029296867),
partition o_ordk3421 values less than (15033691399),
partition o_ordk3422 values less than (15038085930),
partition o_ordk3423 values less than (15042480461),
partition o_ordk3424 values less than (15046874992),
partition o_ordk3425 values less than (15051269524),
partition o_ordk3426 values less than (15055664055),
partition o_ordk3427 values less than (15060058586),
partition o_ordk3428 values less than (15064453117),
partition o_ordk3429 values less than (15068847649),
partition o_ordk3430 values less than (15073242180),
partition o_ordk3431 values less than (15077636711),
partition o_ordk3432 values less than (15082031242),
partition o_ordk3433 values less than (15086425774),
partition o_ordk3434 values less than (15090820305),
partition o_ordk3435 values less than (15095214836),
partition o_ordk3436 values less than (15099609367),
partition o_ordk3437 values less than (15104003899),
partition o_ordk3438 values less than (15108398430),
partition o_ordk3439 values less than (15112792961),
partition o_ordk3440 values less than (15117187492),
partition o_ordk3441 values less than (15121582024),
partition o_ordk3442 values less than (15125976555),
partition o_ordk3443 values less than (15130371086),
partition o_ordk3444 values less than (15134765617),
partition o_ordk3445 values less than (15139160149),
partition o_ordk3446 values less than (15143554680),
partition o_ordk3447 values less than (15147949211),
partition o_ordk3448 values less than (15152343742),
partition o_ordk3449 values less than (15156738274),
partition o_ordk3450 values less than (15161132805),
partition o_ordk3451 values less than (15165527336),
partition o_ordk3452 values less than (15169921867),
partition o_ordk3453 values less than (15174316399),
partition o_ordk3454 values less than (15178710930),
partition o_ordk3455 values less than (15183105461),
partition o_ordk3456 values less than (15187499992),
partition o_ordk3457 values less than (15191894524),
partition o_ordk3458 values less than (15196289055),
partition o_ordk3459 values less than (15200683586),
partition o_ordk3460 values less than (15205078117),
partition o_ordk3461 values less than (15209472649),
partition o_ordk3462 values less than (15213867180),
partition o_ordk3463 values less than (15218261711),
partition o_ordk3464 values less than (15222656242),
partition o_ordk3465 values less than (15227050774),
partition o_ordk3466 values less than (15231445305),
partition o_ordk3467 values less than (15235839836),
partition o_ordk3468 values less than (15240234367),
partition o_ordk3469 values less than (15244628899),
partition o_ordk3470 values less than (15249023430),
partition o_ordk3471 values less than (15253417961),
partition o_ordk3472 values less than (15257812492),
partition o_ordk3473 values less than (15262207024),
partition o_ordk3474 values less than (15266601555),
partition o_ordk3475 values less than (15270996086),
partition o_ordk3476 values less than (15275390617),
partition o_ordk3477 values less than (15279785149),
partition o_ordk3478 values less than (15284179680),
partition o_ordk3479 values less than (15288574211),
partition o_ordk3480 values less than (15292968742),
partition o_ordk3481 values less than (15297363274),
partition o_ordk3482 values less than (15301757805),
```

```
partition o_ordk3483 values less than (15306152336),
partition o_ordk3484 values less than (15310546867),
partition o_ordk3485 values less than (15314941399),
partition o_ordk3486 values less than (15319335930),
partition o_ordk3487 values less than (15323730461),
partition o_ordk3488 values less than (15328124992),
partition o_ordk3489 values less than (15332519524),
partition o_ordk3490 values less than (15336914055),
partition o_ordk3491 values less than (15341308586),
partition o_ordk3492 values less than (15345703117),
partition o_ordk3493 values less than (15350097649),
partition o_ordk3494 values less than (15354492180),
partition o_ordk3495 values less than (15358886711),
partition o_ordk3496 values less than (15363281242),
partition o_ordk3497 values less than (15367675774),
partition o_ordk3498 values less than (15372070305),
partition o_ordk3499 values less than (15376464836),
partition o_ordk3500 values less than (15380859367),
partition o_ordk3501 values less than (15385253899),
partition o_ordk3502 values less than (15389648430),
partition o_ordk3503 values less than (15394042961),
partition o_ordk3504 values less than (15398437492),
partition o_ordk3505 values less than (15402832024),
partition o_ordk3506 values less than (15407226555),
partition o_ordk3507 values less than (15411621086),
partition o_ordk3508 values less than (15416015617),
partition o_ordk3509 values less than (15420410149),
partition o_ordk3510 values less than (15424804680),
partition o_ordk3511 values less than (15429199211),
partition o_ordk3512 values less than (15433593742),
partition o_ordk3513 values less than (15437988274),
partition o_ordk3514 values less than (15442382805),
partition o_ordk3515 values less than (15446777336),
partition o_ordk3516 values less than (15451171867),
partition o_ordk3517 values less than (15455566399),
partition o_ordk3518 values less than (15459960930),
partition o_ordk3519 values less than (15464355461),
partition o_ordk3520 values less than (15468749992),
partition o_ordk3521 values less than (15473144524),
partition o_ordk3522 values less than (15477539055),
partition o_ordk3523 values less than (15481933586),
partition o_ordk3524 values less than (15486328117),
partition o_ordk3525 values less than (15490722649),
partition o_ordk3526 values less than (15495117180),
partition o_ordk3527 values less than (15499511711),
partition o_ordk3528 values less than (15503906242),
partition o_ordk3529 values less than (15508300773),
partition o_ordk3530 values less than (15512695305),
partition o_ordk3531 values less than (15517089836),
partition o_ordk3532 values less than (15521484367),
partition o_ordk3533 values less than (15525878898),
partition o_ordk3534 values less than (15530273430),
partition o_ordk3535 values less than (15534667961),
partition o_ordk3536 values less than (15539062492),
partition o_ordk3537 values less than (15543457023),
partition o_ordk3538 values less than (15547851555),
partition o_ordk3539 values less than (15552246086),
partition o_ordk3540 values less than (15556640617),
partition o_ordk3541 values less than (15561035148),
partition o_ordk3542 values less than (15565429680),
partition o_ordk3543 values less than (15569824211),
partition o_ordk3544 values less than (15574218742),
partition o_ordk3545 values less than (15578613273),
partition o_ordk3546 values less than (15583007805),
partition o_ordk3547 values less than (15587402336),
partition o_ordk3548 values less than (15591796867),
partition o_ordk3549 values less than (15596191398),
partition o_ordk3550 values less than (15600585930),
partition o_ordk3551 values less than (15604980461),
partition o_ordk3552 values less than (15609374992),
partition o_ordk3553 values less than (15613769523),
partition o_ordk3554 values less than (15618164055),
partition o_ordk3555 values less than (15622558586),
partition o_ordk3556 values less than (15626953117),
partition o_ordk3557 values less than (15631347648),
partition o_ordk3558 values less than (15635742180),
partition o_ordk3559 values less than (15640136711),
partition o_ordk3560 values less than (15644531242),
partition o_ordk3561 values less than (15648925773),
partition o_ordk3562 values less than (15653320305),
partition o_ordk3563 values less than (15657714836),
partition o_ordk3564 values less than (15662109367),
partition o_ordk3565 values less than (15666503898),
partition o_ordk3566 values less than (15670898430),
partition o_ordk3567 values less than (15675292961),
partition o_ordk3568 values less than (15679687492),
partition o_ordk3569 values less than (15684082023),
partition o_ordk3570 values less than (15688476555),
partition o_ordk3571 values less than (15692871086),
partition o_ordk3572 values less than (15697265617),
partition o_ordk3573 values less than (15701660148),
partition o_ordk3574 values less than (15706054680),
partition o_ordk3575 values less than (15710449211),
partition o_ordk3576 values less than (15714843742),
partition o_ordk3577 values less than (15719238273),
partition o_ordk3578 values less than (15723632805),
partition o_ordk3579 values less than (15728027336),
partition o_ordk3580 values less than (15732421867),
partition o_ordk3581 values less than (15736816398),
partition o_ordk3582 values less than (15741210930),
partition o_ordk3583 values less than (15745605461),
partition o_ordk3584 values less than (15749999992),
partition o_ordk3585 values less than (15754394523),
partition o_ordk3586 values less than (15758789055),
partition o_ordk3587 values less than (15763183586),
partition o_ordk3588 values less than (15767578117),
partition o_ordk3589 values less than (15771972648),
partition o_ordk3590 values less than (15776367180),
partition o_ordk3591 values less than (15780761711),
partition o_ordk3592 values less than (15785156242),
partition o_ordk3593 values less than (15789550773),
partition o_ordk3594 values less than (15793945305),
partition o_ordk3595 values less than (15798339836),
partition o_ordk3596 values less than (15802734367),
partition o_ordk3597 values less than (15807128898),
partition o_ordk3598 values less than (15811523430),
partition o_ordk3599 values less than (15815917961),
partition o_ordk3600 values less than (15820312492),
partition o_ordk3601 values less than (15824707023),
partition o_ordk3602 values less than (15829101555),
partition o_ordk3603 values less than (15833496086),
partition o_ordk3604 values less than (15837890617),
partition o_ordk3605 values less than (15842285148),
partition o_ordk3606 values less than (15846679680),
partition o_ordk3607 values less than (15851074211),
partition o_ordk3608 values less than (15855468742),
partition o_ordk3609 values less than (15859863273),
partition o_ordk3610 values less than (15864257805),
partition o_ordk3611 values less than (15868652336),
partition o_ordk3612 values less than (15873046867),
partition o_ordk3613 values less than (15877441398),
partition o_ordk3614 values less than (15881835930),
partition o_ordk3615 values less than (15886230461),
partition o_ordk3616 values less than (15890624992),
partition o_ordk3617 values less than (15895019523),
partition o_ordk3618 values less than (15899414055),
partition o_ordk3619 values less than (15903808586),
partition o_ordk3620 values less than (15908203117),
partition o_ordk3621 values less than (15912597648),
partition o_ordk3622 values less than (15916992180),
partition o_ordk3623 values less than (15921386711),
partition o_ordk3624 values less than (15925781242),
partition o_ordk3625 values less than (15930175773),
partition o_ordk3626 values less than (15934570305),
partition o_ordk3627 values less than (15938964836),
partition o_ordk3628 values less than (15943359367),
partition o_ordk3629 values less than (15947753898),
partition o_ordk3630 values less than (15952148430),
partition o_ordk3631 values less than (15956542961),
partition o_ordk3632 values less than (15960937492),
partition o_ordk3633 values less than (15965332023),
partition o_ordk3634 values less than (15969726555),
partition o_ordk3635 values less than (15974121086),
partition o_ordk3636 values less than (15978515617),
partition o_ordk3637 values less than (15982910148),
partition o_ordk3638 values less than (15987304680),
partition o_ordk3639 values less than (15991699211),
partition o_ordk3640 values less than (15996093742),
partition o_ordk3641 values less than (16000488273),
partition o_ordk3642 values less than (16004882804),
partition o_ordk3643 values less than (16009277336),
partition o_ordk3644 values less than (16013671867),
partition o_ordk3645 values less than (16018066398),
partition o_ordk3646 values less than (16022460929),
partition o_ordk3647 values less than (16026855461),
partition o_ordk3648 values less than (16031249992),
partition o_ordk3649 values less than (16035644523),
partition o_ordk3650 values less than (16040039054),
partition o_ordk3651 values less than (16044433586),
partition o_ordk3652 values less than (16048828117),
partition o_ordk3653 values less than (16053222648),
partition o_ordk3654 values less than (16057617179),
partition o_ordk3655 values less than (16062011711),
partition o_ordk3656 values less than (16066406242),
partition o_ordk3657 values less than (16070800773),
partition o_ordk3658 values less than (16075195304),
partition o_ordk3659 values less than (16079589836),
partition o_ordk3660 values less than (16083984367),
partition o_ordk3661 values less than (16088378898),
partition o_ordk3662 values less than (16092773429),
```

```
partition o_ordk3663 values less than (16097167961),
partition o_ordk3664 values less than (16101562492),
partition o_ordk3665 values less than (16105957023),
partition o_ordk3666 values less than (16110351554),
partition o_ordk3667 values less than (16114746086),
partition o_ordk3668 values less than (16119140617),
partition o_ordk3669 values less than (16123535148),
partition o_ordk3670 values less than (16127929679),
partition o_ordk3671 values less than (16132324211),
partition o_ordk3672 values less than (16136718742),
partition o_ordk3673 values less than (16141113273),
partition o_ordk3674 values less than (16145507804),
partition o_ordk3675 values less than (16149902336),
partition o_ordk3676 values less than (16154296867),
partition o_ordk3677 values less than (16158691398),
partition o_ordk3678 values less than (16163085929),
partition o_ordk3679 values less than (16167480461),
partition o_ordk3680 values less than (16171874992),
partition o_ordk3681 values less than (16176269523),
partition o_ordk3682 values less than (16180664054),
partition o_ordk3683 values less than (16185058586),
partition o_ordk3684 values less than (16189453117),
partition o_ordk3685 values less than (16193847648),
partition o_ordk3686 values less than (16198242179),
partition o_ordk3687 values less than (16202636711),
partition o_ordk3688 values less than (16207031242),
partition o_ordk3689 values less than (16211425773),
partition o_ordk3690 values less than (16215820304),
partition o_ordk3691 values less than (16220214836),
partition o_ordk3692 values less than (16224609367),
partition o_ordk3693 values less than (16229003898),
partition o_ordk3694 values less than (16233398429),
partition o_ordk3695 values less than (16237792961),
partition o_ordk3696 values less than (16242187492),
partition o_ordk3697 values less than (16246582023),
partition o_ordk3698 values less than (16250976554),
partition o_ordk3699 values less than (16255371086),
partition o_ordk3700 values less than (16259765617),
partition o_ordk3701 values less than (16264160148),
partition o_ordk3702 values less than (16268554679),
partition o_ordk3703 values less than (16272949211),
partition o_ordk3704 values less than (16277343742),
partition o_ordk3705 values less than (16281738273),
partition o_ordk3706 values less than (16286132804),
partition o_ordk3707 values less than (16290527336),
partition o_ordk3708 values less than (16294921867),
partition o_ordk3709 values less than (16299316398),
partition o_ordk3710 values less than (16303710929),
partition o_ordk3711 values less than (16308105461),
partition o_ordk3712 values less than (16312499992),
partition o_ordk3713 values less than (16316894523),
partition o_ordk3714 values less than (16321289054),
partition o_ordk3715 values less than (16325683586),
partition o_ordk3716 values less than (16330078117),
partition o_ordk3717 values less than (16334472648),
partition o_ordk3718 values less than (16338867179),
partition o_ordk3719 values less than (16343261711),
partition o_ordk3720 values less than (16347656242),
partition o_ordk3721 values less than (16352050773),
partition o_ordk3722 values less than (16356445304),
partition o_ordk3723 values less than (16360839836),
partition o_ordk3724 values less than (16365234367),
partition o_ordk3725 values less than (16369628898),
partition o_ordk3726 values less than (16374023429),
partition o_ordk3727 values less than (16378417961),
partition o_ordk3728 values less than (16382812492),
partition o_ordk3729 values less than (16387207023),
partition o_ordk3730 values less than (16391601554),
partition o_ordk3731 values less than (16395996086),
partition o_ordk3732 values less than (16400390617),
partition o_ordk3733 values less than (16404785148),
partition o_ordk3734 values less than (16409179679),
partition o_ordk3735 values less than (16413574211),
partition o_ordk3736 values less than (16417968742),
partition o_ordk3737 values less than (16422363273),
partition o_ordk3738 values less than (16426757804),
partition o_ordk3739 values less than (16431152336),
partition o_ordk3740 values less than (16435546867),
partition o_ordk3741 values less than (16439941398),
partition o_ordk3742 values less than (16444335929),
partition o_ordk3743 values less than (16448730461),
partition o_ordk3744 values less than (16453124992),
partition o_ordk3745 values less than (16457519523),
partition o_ordk3746 values less than (16461914054),
partition o_ordk3747 values less than (16466308586),
partition o_ordk3748 values less than (16470703117),
partition o_ordk3749 values less than (16475097648),
partition o_ordk3750 values less than (16479492179),
partition o_ordk3751 values less than (16483886711),
partition o_ordk3752 values less than (16488281242),
partition o_ordk3753 values less than (16492675773),
partition o_ordk3754 values less than (16497070304),
partition o_ordk3755 values less than (16501464835),
partition o_ordk3756 values less than (16505859367),
partition o_ordk3757 values less than (16510253898),
partition o_ordk3758 values less than (16514648429),
partition o_ordk3759 values less than (16519042960),
partition o_ordk3760 values less than (16523437492),
partition o_ordk3761 values less than (16527832023),
partition o_ordk3762 values less than (16532226554),
partition o_ordk3763 values less than (16536621085),
partition o_ordk3764 values less than (16541015617),
partition o_ordk3765 values less than (16545410148),
partition o_ordk3766 values less than (16549804679),
partition o_ordk3767 values less than (16554199210),
partition o_ordk3768 values less than (16558593742),
partition o_ordk3769 values less than (16562988273),
partition o_ordk3770 values less than (16567382804),
partition o_ordk3771 values less than (16571777335),
partition o_ordk3772 values less than (16576171867),
partition o_ordk3773 values less than (16580566398),
partition o_ordk3774 values less than (16584960929),
partition o_ordk3775 values less than (16589355460),
partition o_ordk3776 values less than (16593749992),
partition o_ordk3777 values less than (16598144523),
partition o_ordk3778 values less than (16602539054),
partition o_ordk3779 values less than (16606933585),
partition o_ordk3780 values less than (16611328117),
partition o_ordk3781 values less than (16615722648),
partition o_ordk3782 values less than (16620117179),
partition o_ordk3783 values less than (16624511710),
partition o_ordk3784 values less than (16628906242),
partition o_ordk3785 values less than (16633300773),
partition o_ordk3786 values less than (16637695304),
partition o_ordk3787 values less than (16642089835),
partition o_ordk3788 values less than (16646484367),
partition o_ordk3789 values less than (16650878898),
partition o_ordk3790 values less than (16655273429),
partition o_ordk3791 values less than (16659667960),
partition o_ordk3792 values less than (16664062492),
partition o_ordk3793 values less than (16668457023),
partition o_ordk3794 values less than (16672851554),
partition o_ordk3795 values less than (16677246085),
partition o_ordk3796 values less than (16681640617),
partition o_ordk3797 values less than (16686035148),
partition o_ordk3798 values less than (16690429679),
partition o_ordk3799 values less than (16694824210),
partition o_ordk3800 values less than (16699218742),
partition o_ordk3801 values less than (16703613273),
partition o_ordk3802 values less than (16708007804),
partition o_ordk3803 values less than (16712402335),
partition o_ordk3804 values less than (16716796867),
partition o_ordk3805 values less than (16721191398),
partition o_ordk3806 values less than (16725585929),
partition o_ordk3807 values less than (16729980460),
partition o_ordk3808 values less than (16734374992),
partition o_ordk3809 values less than (16738769523),
partition o_ordk3810 values less than (16743164054),
partition o_ordk3811 values less than (16747558585),
partition o_ordk3812 values less than (16751953117),
partition o_ordk3813 values less than (16756347648),
partition o_ordk3814 values less than (16760742179),
partition o_ordk3815 values less than (16765136710),
partition o_ordk3816 values less than (16769531242),
partition o_ordk3817 values less than (16773925773),
partition o_ordk3818 values less than (16778320304),
partition o_ordk3819 values less than (16782714835),
partition o_ordk3820 values less than (16787109367),
partition o_ordk3821 values less than (16791503898),
partition o_ordk3822 values less than (16795898429),
partition o_ordk3823 values less than (16800292960),
partition o_ordk3824 values less than (16804687492),
partition o_ordk3825 values less than (16809082023),
partition o_ordk3826 values less than (16813476554),
partition o_ordk3827 values less than (16817871085),
partition o_ordk3828 values less than (16822265617),
partition o_ordk3829 values less than (16826660148),
partition o_ordk3830 values less than (16831054679),
partition o_ordk3831 values less than (16835449210),
partition o_ordk3832 values less than (16839843742),
partition o_ordk3833 values less than (16844238273),
partition o_ordk3834 values less than (16848632804),
partition o_ordk3835 values less than (16853027335),
partition o_ordk3836 values less than (16857421867),
partition o_ordk3837 values less than (16861816398),
partition o_ordk3838 values less than (16866210929),
partition o_ordk3839 values less than (16870605460),
partition o_ordk3840 values less than (16874999992),
partition o_ordk3841 values less than (16879394523),
partition o_ordk3842 values less than (16883789054),
```

```
partition o_ordk3843 values less than (16888183585),
partition o_ordk3844 values less than (16892578117),
partition o_ordk3845 values less than (16896972648),
partition o_ordk3846 values less than (16901367179),
partition o_ordk3847 values less than (16905761710),
partition o_ordk3848 values less than (16910156242),
partition o_ordk3849 values less than (16914550773),
partition o_ordk3850 values less than (16918945304),
partition o_ordk3851 values less than (16923339835),
partition o_ordk3852 values less than (16927734367),
partition o_ordk3853 values less than (16932128898),
partition o_ordk3854 values less than (16936523429),
partition o_ordk3855 values less than (16940917960),
partition o_ordk3856 values less than (16945312492),
partition o_ordk3857 values less than (16949707023),
partition o_ordk3858 values less than (16954101554),
partition o_ordk3859 values less than (16958496085),
partition o_ordk3860 values less than (16962890617),
partition o_ordk3861 values less than (16967285148),
partition o_ordk3862 values less than (16971679679),
partition o_ordk3863 values less than (16976074210),
partition o_ordk3864 values less than (16980468742),
partition o_ordk3865 values less than (16984863273),
partition o_ordk3866 values less than (16989257804),
partition o_ordk3867 values less than (16993652335),
partition o_ordk3868 values less than (16998046867),
partition o_ordk3869 values less than (17002441398),
partition o_ordk3870 values less than (17006835929),
partition o_ordk3871 values less than (17011230460),
partition o_ordk3872 values less than (17015624991),
partition o_ordk3873 values less than (17020019523),
partition o_ordk3874 values less than (17024414054),
partition o_ordk3875 values less than (17028808585),
partition o_ordk3876 values less than (17033203116),
partition o_ordk3877 values less than (17037597648),
partition o_ordk3878 values less than (17041992179),
partition o_ordk3879 values less than (17046386710),
partition o_ordk3880 values less than (17050781241),
partition o_ordk3881 values less than (17055175773),
partition o_ordk3882 values less than (17059570304),
partition o_ordk3883 values less than (17063964835),
partition o_ordk3884 values less than (17068359366),
partition o_ordk3885 values less than (17072753898),
partition o_ordk3886 values less than (17077148429),
partition o_ordk3887 values less than (17081542960),
partition o_ordk3888 values less than (17085937491),
partition o_ordk3889 values less than (17090332023),
partition o_ordk3890 values less than (17094726554),
partition o_ordk3891 values less than (17099121085),
partition o_ordk3892 values less than (17103515616),
partition o_ordk3893 values less than (17107910148),
partition o_ordk3894 values less than (17112304679),
partition o_ordk3895 values less than (17116699210),
partition o_ordk3896 values less than (17121093741),
partition o_ordk3897 values less than (17125488273),
partition o_ordk3898 values less than (17129882804),
partition o_ordk3899 values less than (17134277335),
partition o_ordk3900 values less than (17138671866),
partition o_ordk3901 values less than (17143066398),
partition o_ordk3902 values less than (17147460929),
partition o_ordk3903 values less than (17151855460),
partition o_ordk3904 values less than (17156249991),
partition o_ordk3905 values less than (17160644523),
partition o_ordk3906 values less than (17165039054),
partition o_ordk3907 values less than (17169433585),
partition o_ordk3908 values less than (17173828116),
partition o_ordk3909 values less than (17178222648),
partition o_ordk3910 values less than (17182617179),
partition o_ordk3911 values less than (17187011710),
partition o_ordk3912 values less than (17191406241),
partition o_ordk3913 values less than (17195800773),
partition o_ordk3914 values less than (17200195304),
partition o_ordk3915 values less than (17204589835),
partition o_ordk3916 values less than (17208984366),
partition o_ordk3917 values less than (17213378898),
partition o_ordk3918 values less than (17217773429),
partition o_ordk3919 values less than (17222167960),
partition o_ordk3920 values less than (17226562491),
partition o_ordk3921 values less than (17230957023),
partition o_ordk3922 values less than (17235351554),
partition o_ordk3923 values less than (17239746085),
partition o_ordk3924 values less than (17244140616),
partition o_ordk3925 values less than (17248535148),
partition o_ordk3926 values less than (17252929679),
partition o_ordk3927 values less than (17257324210),
partition o_ordk3928 values less than (17261718741),
partition o_ordk3929 values less than (17266113273),
partition o_ordk3930 values less than (17270507804),
partition o_ordk3931 values less than (17274902335),
partition o_ordk3932 values less than (17279296866),
partition o_ordk3933 values less than (17283691398),
partition o_ordk3934 values less than (17288085929),
partition o_ordk3935 values less than (17292480460),
partition o_ordk3936 values less than (17296874991),
partition o_ordk3937 values less than (17301269523),
partition o_ordk3938 values less than (17305664054),
partition o_ordk3939 values less than (17310058585),
partition o_ordk3940 values less than (17314453116),
partition o_ordk3941 values less than (17318847648),
partition o_ordk3942 values less than (17323242179),
partition o_ordk3943 values less than (17327636710),
partition o_ordk3944 values less than (17332031241),
partition o_ordk3945 values less than (17336425773),
partition o_ordk3946 values less than (17340820304),
partition o_ordk3947 values less than (17345214835),
partition o_ordk3948 values less than (17349609366),
partition o_ordk3949 values less than (17354003898),
partition o_ordk3950 values less than (17358398429),
partition o_ordk3951 values less than (17362792960),
partition o_ordk3952 values less than (17367187491),
partition o_ordk3953 values less than (17371582023),
partition o_ordk3954 values less than (17375976554),
partition o_ordk3955 values less than (17380371085),
partition o_ordk3956 values less than (17384765616),
partition o_ordk3957 values less than (17389160148),
partition o_ordk3958 values less than (17393554679),
partition o_ordk3959 values less than (17397949210),
partition o_ordk3960 values less than (17402343741),
partition o_ordk3961 values less than (17406738273),
partition o_ordk3962 values less than (17411132804),
partition o_ordk3963 values less than (17415527335),
partition o_ordk3964 values less than (17419921866),
partition o_ordk3965 values less than (17424316398),
partition o_ordk3966 values less than (17428710929),
partition o_ordk3967 values less than (17433105460),
partition o_ordk3968 values less than (17437499991),
partition o_ordk3969 values less than (17441894523),
partition o_ordk3970 values less than (17446289054),
partition o_ordk3971 values less than (17450683585),
partition o_ordk3972 values less than (17455078116),
partition o_ordk3973 values less than (17459472648),
partition o_ordk3974 values less than (17463867179),
partition o_ordk3975 values less than (17468261710),
partition o_ordk3976 values less than (17472656241),
partition o_ordk3977 values less than (17477050773),
partition o_ordk3978 values less than (17481445304),
partition o_ordk3979 values less than (17485839835),
partition o_ordk3980 values less than (17490234366),
partition o_ordk3981 values less than (17494628898),
partition o_ordk3982 values less than (17499023429),
partition o_ordk3983 values less than (17503417960),
partition o_ordk3984 values less than (17507812491),
partition o_ordk3985 values less than (17512207022),
partition o_ordk3986 values less than (17516601554),
partition o_ordk3987 values less than (17520996085),
partition o_ordk3988 values less than (17525390616),
partition o_ordk3989 values less than (17529785147),
partition o_ordk3990 values less than (17534179679),
partition o_ordk3991 values less than (17538574210),
partition o_ordk3992 values less than (17542968741),
partition o_ordk3993 values less than (17547363272),
partition o_ordk3994 values less than (17551757804),
partition o_ordk3995 values less than (17556152335),
partition o_ordk3996 values less than (17560546866),
partition o_ordk3997 values less than (17564941397),
partition o_ordk3998 values less than (17569335929),
partition o_ordk3999 values less than (17573730460),
partition o_ordk4000 values less than (17578124991),
partition o_ordk4001 values less than (17582519522),
partition o_ordk4002 values less than (17586914054),
partition o_ordk4003 values less than (17591308585),
partition o_ordk4004 values less than (17595703116),
partition o_ordk4005 values less than (17600097647),
partition o_ordk4006 values less than (17604492179),
partition o_ordk4007 values less than (17608886710),
partition o_ordk4008 values less than (17613281241),
partition o_ordk4009 values less than (17617675772),
partition o_ordk4010 values less than (17622070304),
partition o_ordk4011 values less than (17626464835),
partition o_ordk4012 values less than (17630859366),
partition o_ordk4013 values less than (17635253897),
partition o_ordk4014 values less than (17639648429),
partition o_ordk4015 values less than (17644042960),
partition o_ordk4016 values less than (17648437491),
partition o_ordk4017 values less than (17652832022),
partition o_ordk4018 values less than (17657226554),
partition o_ordk4019 values less than (17661621085),
partition o_ordk4020 values less than (17666015616),
partition o_ordk4021 values less than (17670410147),
partition o_ordk4022 values less than (17674804679),
```

```
partition o_ordk4023 values less than (17679199210),
partition o_ordk4024 values less than (17683593741),
partition o_ordk4025 values less than (17687988272),
partition o_ordk4026 values less than (17692382804),
partition o_ordk4027 values less than (17696777335),
partition o_ordk4028 values less than (17701171866),
partition o_ordk4029 values less than (17705566397),
partition o_ordk4030 values less than (17709960929),
partition o_ordk4031 values less than (17714355460),
partition o_ordk4032 values less than (17718749991),
partition o_ordk4033 values less than (17723144522),
partition o_ordk4034 values less than (17727539054),
partition o_ordk4035 values less than (17731933585),
partition o_ordk4036 values less than (17736328116),
partition o_ordk4037 values less than (17740722647),
partition o_ordk4038 values less than (17745117179),
partition o_ordk4039 values less than (17749511710),
partition o_ordk4040 values less than (17753906241),
partition o_ordk4041 values less than (17758300772),
partition o_ordk4042 values less than (17762695304),
partition o_ordk4043 values less than (17767089835),
partition o_ordk4044 values less than (17771484366),
partition o_ordk4045 values less than (17775878897),
partition o_ordk4046 values less than (17780273429),
partition o_ordk4047 values less than (17784667960),
partition o_ordk4048 values less than (17789062491),
partition o_ordk4049 values less than (17793457022),
partition o_ordk4050 values less than (17797851554),
partition o_ordk4051 values less than (17802246085),
partition o_ordk4052 values less than (17806640616),
partition o_ordk4053 values less than (17811035147),
partition o_ordk4054 values less than (17815429679),
partition o_ordk4055 values less than (17819824210),
partition o_ordk4056 values less than (17824218741),
partition o_ordk4057 values less than (17828613272),
partition o_ordk4058 values less than (17833007804),
partition o_ordk4059 values less than (17837402335),
partition o_ordk4060 values less than (17841796866),
partition o_ordk4061 values less than (17846191397),
partition o_ordk4062 values less than (17850585929),
partition o_ordk4063 values less than (17854980460),
partition o_ordk4064 values less than (17859374991),
partition o_ordk4065 values less than (17863769522),
partition o_ordk4066 values less than (17868164054),
partition o_ordk4067 values less than (17872558585),
partition o_ordk4068 values less than (17876953116),
partition o_ordk4069 values less than (17881347647),
partition o_ordk4070 values less than (17885742179),
partition o_ordk4071 values less than (17890136710),
partition o_ordk4072 values less than (17894531241),
partition o_ordk4073 values less than (17898925772),
partition o_ordk4074 values less than (17903320304),
partition o_ordk4075 values less than (17907714835),
partition o_ordk4076 values less than (17912109366),
partition o_ordk4077 values less than (17916503897),
partition o_ordk4078 values less than (17920898429),
partition o_ordk4079 values less than (17925292960),
partition o_ordk4080 values less than (17929687491),
partition o_ordk4081 values less than (17934082022),
partition o_ordk4082 values less than (17938476554),
partition o_ordk4083 values less than (17942871085),
partition o_ordk4084 values less than (17947265616),
partition o_ordk4085 values less than (17951660147),
partition o_ordk4086 values less than (17956054679),
partition o_ordk4087 values less than (17960449210),
partition o_ordk4088 values less than (17964843741),
partition o_ordk4089 values less than (17969238272),
partition o_ordk4090 values less than (17973632804),
partition o_ordk4091 values less than (17978027335),
partition o_ordk4092 values less than (17982421866),
partition o_ordk4093 values less than (17986816397),
partition o_ordk4094 values less than (17991210929),
partition o_ordk4095 values less than (17995605460),
partition o_ordk4096 values less than (MAXVALUE)
);
}
*sql
{
connect tpcd/tpcd;
drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_i_ccustkey
storage (freelists 99 freelist groups 2)
parallel
;
}
```

```
*wait
*bgoff
%e-ixcre
=====================================================
```

3tera_ixcre2.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load1
\sqlldr {}

*load2
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-ixcre
*bgon=1
#####################################################
######################
# Index Creation Phase
*sql
{
connect tpcd/tpcd;
drop index i_ps_partkey_suppkey;
```

```
create unique index i_ps_partkey_suppkey
on partsupp (ps_partkey,ps_suppkey)
pctfree 2
initrans 10
compute statistics
tablespace ts_i_ps
storage (freelists 99 freelist groups 2)
parallel
;
}
*wait
*bgoff
%e-ixcre
```

3tera_anlyz.dat

```
%e-preproc
%b-anlyz
*bgon=4
#######################################################
######################
# Analyze Phase
*sql
{
connect tpcd/tpcd;
execute dbms_stats.gather_schema_stats('tpcd' ,
estimate_percent => 1 , degree => 256 , granularity =>
'GLOBAL' , block_sample => true);
}
*wait
*wait
*bgoff
%e-anlyz
```

3tera_preproc.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal;
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\{}
\select to_char(sysdate, 'MM-DD-YYYY HH24:MI:SS') now
from dual;
\exit;
\!


*load3
\echo '#!/bin/csh' >
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_HOME
/export/home/oracle/oracle817' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv ORACLE_SID  inst2' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv PATH
${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/bin'
>>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo 'setenv LD_LIBRARY_PATH
${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME}/lib:"$LD_LIBRA
RY_PATH"' >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\echo "sqlldr {}" >>
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\chmod a+x
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh
\rcp
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh rep2:/export/home/oracle/kit/load/scripts/
\rsh -n rep2
/export/home/oracle/kit/load/scripts/load1_*getenv(BUM
PX_CTR).sh



*load1
\sqlldr {}

*load2
\sqlldr {}


*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}
```

DBGEN Modifications

driver.c

```
/* @(#)driver.c       2.1.8.4 */
/* main driver for dss banchmark */

#define DECLARER                                /* EXTERN
references get defined here */
#define NO_FUNC (int (*) ()) NULL     /* to clean up
tdefs */
#define NO_LFUNC (long (*) ()) NULL           /* to
clean up tdefs */

#include "config.h"
#include <stdlib.h>
#if (defined(_POSIX_)||!defined(WIN32))
        /* Change for Windows NT */
#include <unistd.h>
#include <sys/wait.h>
#endif /* WIN32 */
#include <stdio.h>                              /* */
#include <limits.h>
#include <math.h>
#include <ctype.h>
#include <signal.h>
#include <string.h>
#include <errno.h>
#ifdef HP
#include <strings.h>
#endif
#if (defined(WIN32)&&!defined(_POSIX_))
#include <process.h>
#pragma warning(disable:4201)
#pragma warning(disable:4214)
#pragma warning(disable:4514)
#define WIN32_LEAN_AND_MEAN
#define NOATOM
#define NOGDICAPMASKS
#define NOMETAFILE
#define NOMINMAX
#define NOMSG
#define NOOPENFILE
#define NORASTEROPS
#define NOSCROLL
#define NOSOUND
#define NOSYSMETRICS
#define NOTEXTMETRIC
#define NOWH
#define NOCOMM
#define NOKANJI
#define NOMCX
#include <windows.h>
#pragma warning(default:4201)
#pragma warning(default:4214)
#endif

#include "dss.h"
#include "dsstypes.h"
#include "bcd2.h"

/*
* Function prototypes
*/
void    usage (void);
int             prep_direct (char *);
int             close_direct (void);
void    kill_load (void);
```

```
int             pload (int tbl);
void    gen_tbl (int tnum, DSS_HUGE start, long count,
long upd_num);
int             pr_drange (int tbl, long min, long cnt,
long num);
int             set_files (int t, int pload);
int             partial (int, int);

#ifdef ORACLE
#include "part.h"
int partition = 0;      /* partition flag       */
vals *values;
FILE *in;               /* fdes for input file  */
FILE  **out;              /* output fd   array    */
FILE *par;              /* fdes for params file */
#ifdef DEBUG
/* for debugging */
FILE *debug_fd = NULL;  /* fd for debugging file */
char debug_fn[FLEN];    /* debug file name       */
extern long row_count;
extern long *part_row;
int j;
#endif /* DEBUG */
char param[FLEN];
extern char outname[];
extern FILE* outparam;    /* fd for output parameter
file                 */
int use_outparam = 0;   /* flag to designate whether
to use outparam file */
long numpart = 0;       /* number of partitions
*/
long partnum;           /* partition number
*/
int numcol = 0;         /* number of columns
*/
char delim = '|';       /* flatfile delimiter
*/
#endif /* ORACLE */

extern int optind, opterr;
extern char *optarg;
long rowcnt = 0, minrow = 0, upd_num = 0;
double flt_scale;
#if (defined(WIN32)&&!defined(_POSIX_))
char *spawn_args[25];
#endif



/*
* general table descriptions. See dss.h for details on
structure
* NOTE: tables with no scaling info are scaled
according to
* another table
*
*
* the following is based on the tdef structure defined
in dss.h as:
* typedef struct
* {
* char     *name;            -- name of the table;
*                               flat file output in
<name>.tbl
* long      base;            -- base scale rowcount of
table;
*                               0 if derived
* int       (*header) ();    -- function to prep
output
* int       (*loader[2]) (); -- functions to present
output
* long      (*gen_seed) ();  -- functions to seed the
RNG
* int       (*verify) ();    -- function to verfiy the
data set without building it
* int       child;           -- non-zero if there is
an associated detail table
* unsigned long vtotal;      -- "checksum" total
* }         tdef;
*
*/

/*
* flat file print functions; used with -F(lat) option
*/
int pr_cust (customer_t * c, int mode);
int pr_line (order_t * o, int mode);
int pr_order (order_t * o, int mode);
int pr_part (part_t * p, int mode);
int pr_psupp (part_t * p, int mode);
int pr_supp (supplier_t * s, int mode);
int pr_order_line (order_t * o, int mode);
int pr_part_psupp (part_t * p, int mode);
int pr_nation (code_t * c, int mode);
int pr_region (code_t * c, int mode);

/*
* inline load functions; used with -D(irect) option
*/
int ld_cust (customer_t * c, int mode);
int ld_line (order_t * o, int mode);
int ld_order (order_t * o, int mode);
int ld_part (part_t * p, int mode);
int ld_psupp (part_t * p, int mode);
int ld_supp (supplier_t * s, int mode);
int ld_order_line (order_t * o, int mode);
int ld_part_psupp (part_t * p, int mode);
int ld_nation (code_t * c, int mode);
int ld_region (code_t * c, int mode);

/*
* seed generation functions; used with '-O s' option
*/
long sd_cust (int child, long skip_count);
long sd_line (int child, long skip_count);
long sd_order (int child, long skip_count);
long sd_part (int child, long skip_count);
long sd_psupp (int child, long skip_count);
long sd_supp (int child, long skip_count);
long sd_order_line (int child, long skip_count);
long sd_part_psupp (int child, long skip_count);

/*
* header output functions); used with -h(eader) option
*/
int hd_cust (FILE * f);
int hd_line (FILE * f);
int hd_order (FILE * f);
int hd_part (FILE * f);
int hd_psupp (FILE * f);
int hd_supp (FILE * f);
int hd_order_line (FILE * f);
int hd_part_psupp (FILE * f);
int hd_nation (FILE * f);
int hd_region (FILE * f);

/*
* data verfication functions; used with -O v option
*/
int vrf_cust (customer_t * c, int mode);
int vrf_line (order_t * o, int mode);
int vrf_order (order_t * o, int mode);
int vrf_part (part_t * p, int mode);
int vrf_psupp (part_t * p, int mode);
int vrf_supp (supplier_t * s, int mode);
int vrf_order_line (order_t * o, int mode);
int vrf_part_psupp (part_t * p, int mode);
int vrf_nation (code_t * c, int mode);
int vrf_region (code_t * c, int mode);


tdef tdefs[] =
{
        {"part.tbl", "part table", 200000, hd_part,
                {pr_part, ld_part}, sd_part, vrf_part,
PSUPP, 0},
        {"partsupp.tbl", "partsupplier table", 200000,
hd_psupp,
                {pr_psupp, ld_psupp}, sd_psupp,
vrf_psupp, NONE, 0},
        {"supplier.tbl", "suppliers table", 10000,
hd_supp,
                {pr_supp, ld_supp}, sd_supp, vrf_supp,
NONE, 0},
        {"customer.tbl", "customers table", 150000,
hd_cust,
                {pr_cust, ld_cust}, sd_cust, vrf_cust,
NONE, 0},
        {"orders.tbl", "order table", 150000,
hd_order,
                {pr_order, ld_order}, sd_order,
vrf_order, LINE, 0},
        {"lineitem.tbl", "lineitem table", 150000,
hd_line,
                {pr_line, ld_line}, sd_line, vrf_line,
NONE, 0},
        {"orders.tbl", "orders/lineitem tables",
150000, hd_order_line,
                {pr_order_line, ld_order_line},
sd_order, vrf_order_line, LINE, 0},
```

```
        {"part.tbl", "part/partsupplier tables",
200000, hd_part_psupp,
                {pr_part_psupp, ld_part_psupp},
sd_part, vrf_part_psupp, PSUPP, 0},
        {"nation.tbl", "nation table", NATIONS_MAX,
hd_nation,
                {pr_nation, ld_nation}, NO_LFUNC,
vrf_nation, NONE, 0},
        {"region.tbl", "region table", NATIONS_MAX,
hd_region,
                {pr_region, ld_region}, NO_LFUNC,
vrf_region, NONE, 0},
};

int *pids;


/*
* routines to handle the graceful cleanup of multi-
process loads
*/

void
stop_proc (int signum)
{
        exit (0);
}

void
kill_load (void)
{
        int i;

#if !defined(U2200) && !defined(DOS)
        for (i = 0; i < children; i++)
                if (pids[i])
                        KILL (pids[i]);
#endif /* !U2200 && !DOS */
                return;
}

/*
* re-set default output file names
*/
int
set_files (int i, int pload)
{
        char line[80], *new_name;

        if (table & (1 << i))
child_table:
        {
                if (pload != -1)
                        sprintf (line, "%s.%d",
tdefs[i].name, pload);
                else
                {
                        printf ("Enter new destination
for %s data: ",
                                tdefs[i].name);
                        if (fgets (line, sizeof (line),
stdin) == NULL)
                                return (-1);;
                        if ((new_name = strchr (line,
'\n')) != NULL)
                                *new_name = '\0';
                        if (strlen (line) == 0)
                                return (0);
                }
                new_name = (char *) malloc (strlen
(line) + 1);
                MALLOC_CHECK (new_name);
                strcpy (new_name, line);
                tdefs[i].name = new_name;
                if (tdefs[i].child != NONE)
                {
                        i = tdefs[i].child;
                        tdefs[i].child = NONE;
                        goto child_table;
                }
        }

        return (0);
}



/*
* read the distributions needed in the benchamrk
*/
void
load_dists (void)
{
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"p_cntr", &p_cntr_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"colors", &colors);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"p_types", &p_types_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"nations", &nations);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"regions", &regions);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"o_oprio",
                        &o_priority_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"instruct",
                        &l_instruct_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"smode", &l_smode_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"category",
                        &l_category_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"rflag", &l_rflag_set);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"msegmnt", &c_mseg_set);

        /* load the distributions that contain text
generation */
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"nouns", &nouns);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"verbs", &verbs);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"adjectives", &adjectives);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"adverbs", &adverbs);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"auxillaries", &auxillaries);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"terminators", &terminators);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"articles", &articles);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"prepositions", &prepositions);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"grammar", &grammar);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"np", &np);
        read_dist (env_config (DIST_TAG, DIST_DFLT),
"vp", &vp);

}

/*
* generate a particular table
*/
void
gen_tbl (int tnum, DSS_HUGE start, long count, long
upd_num)
{
        static order_t o;
        supplier_t supp;
        customer_t cust;
        part_t part;
        code_t code;
        static int completed = 0;
        static int init = 0;
        DSS_HUGE i;

        int rows_per_segment=0;
        int rows_this_segment=0;
        int residual_rows=0;

        if (insert_segments)
          {
                rows_per_segment = count /
insert_segments;
                residual_rows = count -
(rows_per_segment * insert_segments);
          }
        if (init == 0)
        {
                INIT_HUGE(o.okey);
                for (i=0; i < O_LCNT_MAX; i++)
                        INIT_HUGE(o.l[i].okey);
                init = 1;
        }
```

```
        for (i = start; count; count--, i++)
        {
                LIFENOISE (1000, i);
                row_start(tnum);

                switch (tnum)
                {
                case LINE:
                case ORDER:
                case ORDER_LINE:
                        mk_order (i, &o, upd_num %
10000);

                  if (insert_segments  && (upd_num >
0))
                        if((upd_num / 10000) <
residual_rows)
                                {
                                if((++rows_this_segment)
> rows_per_segment)
                                        {

                                        rows_this_segmen
t=0;
                                        upd_num +=
10000;
                                        }
                                }
                        else
                                {
                                if((++rows_this_segment)
>= rows_per_segment)
                                        {
                                        rows_this_segmen
t=0;
                                        upd_num +=
10000;
                                        }
                                }

                        if (set_seeds == 0)
                                if (validate)
                                        tdefs[tnum].veri
fy(&o, 0);
                                else
                                        tdefs[tnum].load
er[direct] (&o, upd_num);
                        break;
                case SUPP:
                        mk_supp (i, &supp);
                        if (set_seeds == 0)
                                if (validate)
                                        tdefs[tnum].veri
fy(&supp, 0);
                                else
                                        tdefs[tnum].load
er[direct] (&supp, upd_num);
                        break;
                case CUST:
                        mk_cust (i, &cust);
                        if (set_seeds == 0)
                                if (validate)
                                        tdefs[tnum].veri
fy(&cust, 0);
                                else
                                        tdefs[tnum].load
er[direct] (&cust, upd_num);
                        break;
                case PSUPP:
                case PART:
                case PART_PSUPP:
                        mk_part (i, &part);
                        if (set_seeds == 0)
                                if (validate)
                                        tdefs[tnum].veri
fy(&part, 0);
                                else
                                        tdefs[tnum].load
er[direct] (&part, upd_num);
                        break;
                case NATION:
                        mk_nation (i, &code);
                        if (set_seeds == 0)
                                if (validate)
                                        tdefs[tnum].veri
fy(&code, 0);
                                else
                                        tdefs[tnum].load
er[direct] (&code, 0);
                        break;
                case REGION:
                        mk_region (i, &code);
                        if (set_seeds == 0)
                                if (validate)
                                        tdefs[tnum].veri
fy(&code, 0);
                                else
                                        tdefs[tnum].load
er[direct] (&code, 0);
                        break;
                }
                row_stop(tnum);
                if (set_seeds && (i % tdefs[tnum].base)
< 2)
                {
                        printf("\nSeeds for %s at
rowcount %ld\n", tdefs[tnum].comment, i);
                        dump_seeds(tnum);
                }
        }
        completed |= 1 << tnum;
}



void
usage (void)
{
        fprintf (stderr, "%s\n%s\n\t%s\n%s %s\n\n",
                "USAGE:",
                "dbgen [-{vfFD}] [-O {fhmsv}][-T
{pcsoPSOL}]",
                "[-s <scale>][-C <procs>][-S <step>]",
                "dbgen [-v] [-O {dfhmr}] [-s <scale>]",
                "[-U <updates>] [-r <percent>]");
        fprintf (stderr, "-b <s> -- load distributions
for <s>\n");
        fprintf (stderr, "-C <n> -- use <n> processes
to generate data\n");
        fprintf (stderr, "          [Under DOS, must
be used with -S]\n");
        fprintf (stderr, "-D     -- do database load
in line\n");
    fprintf (stderr, "-d <n> -- split deletes between
<n> files\n");
        fprintf (stderr, "-f     -- force. Overwrite
existing files\n");
        fprintf (stderr, "-F     -- generate flat
files output\n");
        fprintf (stderr, "-h     -- display this
message\n");
    fprintf (stderr, "-i <n> -- split inserts between
<n> files\n");
        fprintf (stderr, "-n <s> -- inline load into
database <s>\n");
        fprintf (stderr, "-O d   -- generate SQL
syntax for deletes\n");
        fprintf (stderr, "-O f   -- over-ride default
output file names\n");
        fprintf (stderr, "-O h   -- output files with
headers\n");
        fprintf (stderr, "-O m   -- produce columnar
output\n");
        fprintf (stderr, "-O r   -- generate key
ranges for deletes.\n");
        fprintf (stderr, "-O v   -- Verify data set
without generating it.\n");
        fprintf (stderr, "-q     -- enable QUIET
mode\n");
        fprintf (stderr, "-r <n> -- updates refresh
(n/100)%% of the\n");
        fprintf (stderr, "          data set\n");
        fprintf (stderr, "-s <n> -- set Scale Factor
(SF) to  <n> \n");
        fprintf (stderr, "-S <n> -- build the <n>th
step of the data/update set\n");
        fprintf (stderr, "-T c   -- generate cutomers
ONLY\n");
        fprintf (stderr, "-T l   -- generate
nation/region ONLY\n");
        fprintf (stderr, "-T L   -- generate lineitem
ONLY\n");
        fprintf (stderr, "-T n   -- generate nation
ONLY\n");
        fprintf (stderr, "-T o   -- generate
orders/lineitem ONLY\n");
        fprintf (stderr, "-T O   -- generate orders
ONLY\n");
        fprintf (stderr, "-T p   -- generate
```

```
parts/partsupp ONLY\n");
        fprintf (stderr, "-T P   -- generate parts
ONLY\n");
        fprintf (stderr, "-T r   -- generate region
ONLY\n");
        fprintf (stderr, "-T s   -- generate suppliers
ONLY\n");
        fprintf (stderr, "-T S   -- generate partsupp
ONLY\n");
        fprintf (stderr, "-U <s> -- generate <s>
update sets\n");
        fprintf (stderr, "-v     -- enable VERBOSE
mode\n");
#ifdef ORACLE
    fprintf(stderr,
"\n=================================\n");
    fprintf(stderr, "Additions for partitioning:\n");
    fprintf(stderr, "-p              -- enables
partitioning\n");
    fprintf(stderr, "-I <param file> -- input
parameter file\n");
    fprintf(stderr, "-a <out list>   -- output prefix
list\n");
    fprintf(stderr, "-o <out prefix> -- output file
prefix, -a option overrides -o option\n");
    fprintf(stderr, "Format of parameter file\n");
    fprintf(stderr, "<delimiter>\n");
    fprintf(stderr, "<number of partitions>\n");
    fprintf(stderr, "<column type>\n");
    fprintf(stderr, "<column pos> - 1 for first
column, 2 for 2nd column, ...\n");
    fprintf(stderr, "<partition val1>\n");
    fprintf(stderr, "<partition val2>\n");
    fprintf(stderr, "...\n");
    fprintf(stderr, "<partition valm>\n");
    fprintf(stderr, "Notes:\n");
    fprintf(stderr, "%s\n",
            "Column types are: 0 for numbers, 1 for
strings, 2 for dates");
    fprintf(stderr,
"=================================\n\n");
#endif /* ORACLE */
        fprintf (stderr,
                "\nTo generate the SF=1 (1GB),
validation database population, use:\n");
        fprintf (stderr, "\tdbgen -vfF -s 1\n");
        fprintf (stderr, "\nTo generate updates for a
SF=1 (1GB), use:\n");
        fprintf (stderr, "\tdbgen -v -U 1 -s 1\n");
}

/*
* pload() -- handle the parallel loading of tables
*/
#ifndef DOS
/*
* int partial(int tbl, int s) -- generate the s-th
part of the named tables data
*/
int
partial (int tbl, int s)
{
        long long rowcnt;
        long extra;

        if (verbose > 0)
        {
                fprintf (stderr, "\tStarting to load
stage %d of %d for %s...",
                         s, children,
tdefs[tbl].comment);
        }

        if (direct == 0)
                set_files (tbl, s);

        rowcnt = (long long)set_state(tbl, scale,
children, s, &extra);

        if (s == children)
                gen_tbl (tbl, rowcnt * (s - 1) + 1,
rowcnt + extra, upd_num);
        else
                gen_tbl (tbl, rowcnt * (s - 1) + 1,
rowcnt, upd_num);

        if (verbose > 0)
                fprintf (stderr, "done.\n");

        return (0);
}


int
pload (int tbl)
{
        int c = 0, i, status;

        if (verbose > 0)
        {
                fprintf (stderr, "Starting %d children
to load %s",
                         children, tdefs[tbl].comment);
        }
        for (c = 0; c < children; c++)
        {
                pids[c] = SPAWN ();
                if (pids[c] == -1)
                {
                        perror ("Child loader not
created");
                        kill_load ();
                        exit (-1);
                }
                else if (pids[c] == 0)        /* CHILD
*/
                {
                        SET_HANDLER (stop_proc);
                        verbose = 0;
                        partial (tbl, c+1);
                        exit (0);
                }
                else if (verbose > 0)
        /* PARENT */
                        fprintf (stderr, ".");
        }

        if (verbose > 0)
                fprintf (stderr, "waiting...");

        c = children;
        while (c)
        {
                i = WAIT (&status, pids[c - 1]);
                if (i == -1 && children)
                {
                        if (errno == ECHILD)
                                fprintf (stderr,
"\nCould not wait on pid %d\n", pids[c - 1]);
                        else if (errno == EINTR)
                                fprintf (stderr,
"\nProcess %d stopped abnormally\n", pids[c - 1]);
                        else if (errno == EINVAL)
                                fprintf (stderr,
"\nProgram bug\n");
                }
                if (! WIFEXITED(status)) {
                        (void) fprintf(stderr,
"\nProcess %d: ", i);
                        if (WIFSIGNALED(status)) {
                                (void) fprintf(stderr,
"rcvd signal %d\n",
                                        WTERMSIG(status)
);
                                } else if
(WIFSTOPPED(status)) {
                                (void) fprintf(stderr,
"stopped, signal %d\n",
                                        WSTOPSIG(status)
);
                                }

                        }
                c--;
        }

        if (verbose > 0)
                fprintf (stderr, "done\n");
        return (0);
}
#endif /* !DOS */

void
process_options (int count, char **vector)
{
        int option;

#ifdef ORACLE
    /* set some default */
    if (partition) {
```

```
      (void) sprintf(param,"%s","./part.param");
      (void) sprintf(outname,"%s","/dev/sqlldr");
    }
#endif /* ORACLE */
#ifndef ORACLE
       while ((option = getopt (count, vector,
              "b:C:Dd:Ffi:hn:O:P:qr:s:S:T:U:v")) !=
-1)
#else /* ORACLE */
       while ((option = getopt (count, vector,
              "b:C:Dd:Ffi:hn:O:P:qr:s:S:T:U:vI:o:a:p"
)) != -1)
#endif /* ORACLE */
       switch (option)
             {
#ifdef ORACLE
           case 'p':
             partition = 1;
             break;
           case 'I':
             (void) strcpy(param,optarg);
             break;
           case 'o':
             strcpy(outname, optarg);
             break;
           case 'a':
             if ((outparam = fopen(optarg, "r")) ==
NULL) {
               fprintf(stderr, "Error opening file
%s\n", optarg);
             }
             use_outparam++;
             break;
#endif /* ORACLE */
             case 'b':
       /* load distributions from named file */
d_path = (char *)malloc(strlen(optarg) + 1);
                    MALLOC_CHECK(d_path);
                    strcpy(d_path, optarg);
                    break;
             case 'q':
       /* all prompts disabled */
                    verbose = -1;
                    break;
             case 'i':
                    insert_segments = atoi
(optarg);
                    break;
             case 'd':
                    delete_segments = atoi
(optarg);
                    break;
         case 'S':                          /*
generate a particular STEP */
             step = atoi (optarg);
             break;
         case 'v':                          /* life
noises enabled */
             verbose = 1;
             break;
         case 'f':                          /* blind
overwrites; Force */
             force = 1;
             break;
         case 'T':                          /*
generate a specifc table */
             switch (*optarg)
             {
             case 'c':                      /*
generate customer ONLY */
                    table = 1 << CUST;
                    break;
             case 'L':                      /*
generate lineitems ONLY */
                    table = 1 << LINE;
                    break;
             case 'l':                      /*
generate code table ONLY */
                    table = 1 << NATION;
                    table |= 1 << REGION;
                    break;
             case 'n':                      /*
generate nation table ONLY */
                    table = 1 << NATION;
                    break;
             case 'O':                      /*
generate orders ONLY */
                    table = 1 << ORDER;
                    break;
             case 'o':                      /*
generate orders/lineitems ONLY */
                    table = 1 << ORDER_LINE;
                    break;
             case 'P':                      /*
generate part ONLY */
                    table = 1 << PART;
                    break;
             case 'p':                      /*
generate part/partsupp ONLY */
                    table = 1 << PART_PSUPP;
                    break;
             case 'r':                      /*
generate region table ONLY */
                    table = 1 << REGION;
                    break;
             case 'S':                      /*
generate partsupp ONLY */
                    table = 1 << PSUPP;
                    break;
             case 's':                      /*
generate suppliers ONLY */
                    table = 1 << SUPP;
                    break;
             default:
                    fprintf (stderr, "Unknown
table name %s\n",
                           optarg);
                    usage ();
                    exit (1);
             }
             break;
             case 's':
       /* scale by Percentage of base rowcount */
             case 'P':
       /* for backward compatibility */
                    flt_scale = atof (optarg);
                    if (flt_scale < MIN_SCALE)
                    {
                           int i;

                           scale = 1;
                           for (i = PART; i <
REGION; i++)
                           {
                                  tdefs[i].base
*= flt_scale;
                                  if
(tdefs[i].base < 1)
tdefs[i].base = 1;
                           }
                    }
                    else
                           scale = (long)
flt_scale;
                    if (scale > MAX_SCALE)
                    {
                           fprintf (stderr, "%s
%5.0f %s\n\t%s\n\n",
                                  "NOTE: Data
generation for scale factors >",
                                  MAX_SCALE,
                                  "GB is still
in development,",
                                  "and is not
yet supported.\n");
                           fprintf (stderr,
                                  "Your
resulting data set MAY NOT BE COMPLIANT!\n");
                    }
                    break;
             case 'O':
       /* optional actions */
                    switch (tolower (*optarg))
                    {
                    case 'd':
       /* generate SQL for deletes */
                           gen_sql = 1;
                           break;
                    case 'f':
       /* over-ride default file names */
                           fnames = 1;
                           break;
                    case 'h':
       /* generate headers */
                           header = 1;
                           break;
                    case 'm':
       /* generate columnar output */
                           columnar = 1;
```

```
                                break;
                        case 'r':
        /* generate key ranges for delete */
                                gen_rng = 1;
                                break;
                        case 's':
        /* calibrate the RNG usage */
                                set_seeds = 1;
                                break;
                        case 'v':
        /* validate the data set */
                                validate = 1;
                                break;
                        default:
                                fprintf (stderr,
"Unknown option name %s\n",
                                        optarg);
                                usage ();
                                exit (1);
                        }
                        break;
                        case 'D':
        /* direct load of generated data */
                                direct = 1;
                                break;
                        case 'F':
        /* generate flat files for later loading */
                                direct = 0;
                                break;
                        case 'U':
        /* generate flat files for update stream */
                                updates = atoi
(optarg);
                                break;
                        case 'r':
        /* set the refresh (update) percentage */
                                refresh = atoi
(optarg);
                                break;
#ifndef DOS
                        case 'C':
                                children = atoi
(optarg);
                                break;
#endif /* !DOS */
                        case 'n':
        /* set name of database for direct load */
                                db_name = (char *)
malloc (strlen (optarg) + 1);
                                MALLOC_CHECK
(db_name);
                                strcpy (db_name,
optarg);
                                break;
                        default:
                                printf ("ERROR: option
'%c' unknown.\n",
*(vector[optind] + 1));
                        case 'h':
        /* something unexpected */
                                fprintf (stderr,
                                        "%s Population
Generator (Version %d.%d.%d%s)\n",
                                        NAME, VERSION,
RELEASE,
                                        MODIFICATION,
PATCH);
                                fprintf (stderr,
"Copyright %s %s\n", TPC, C_DATES);
                                usage ();
                                exit (1);
          }

#ifndef DOS
        if (children != 1 && step == -1)
                {
                pids = malloc(children *
sizeof(pid_t));
                MALLOC_CHECK(pids)
                }
#else
        if (children != 1 && step < 0)
                {
                fprintf(stderr, "ERROR: -C must be
accompanied by -S on this platform\n");
                exit(1);
                }
#endif /* DOS */
        return;
}

/*
* MAIN
*
* assumes the existance of getopt() to clean up the
command
* line handling
*/
int
main (int ac, char **av)
{
        int i;

        table = (1 << CUST) |
                (1 << SUPP) |
                (1 << NATION) |
                (1 << REGION) |
                (1 << PART_PSUPP) |
                (1 << ORDER_LINE);
        force = 0;
    insert_segments=0;
    delete_segments=0;
    insert_orders_segment=0;
    insert_lineitem_segment=0;
    delete_segment=0;
        verbose = 0;
        columnar = 0;
        set_seeds = 0;
        header = 0;
        direct = 0;
        scale = 1;
        flt_scale = 1.0;
        updates = 0;
        refresh = UPD_PCT;
        step = -1;
        tdefs[ORDER].base *=
                ORDERS_PER_CUST;                        /
* have to do this after init */
        tdefs[LINE].base *=
                ORDERS_PER_CUST;                        /
* have to do this after init */
        tdefs[ORDER_LINE].base *=
                ORDERS_PER_CUST;                        /
* have to do this after init */
        fnames = 0;
        db_name = NULL;
        gen_sql = 0;
        gen_rng = 0;
        children = 1;
        d_path = NULL;

#ifdef NO_SUPPORT
        signal (SIGINT, exit);
#endif /* NO_SUPPORT */
        process_options (ac, av);
#ifdef ORACLE
    if (partition) {
      if ((par = fopen(param,"r")) == NULL) {
        fprintf(stderr, "Unable to open file '%s'.\n",
param);
        fprintf(stderr, "%s: %s\n", param,
strerror(errno));
        exit(-1);
      }
      (void) process_param_file();
    }
#endif /* ORACLE */
#if (defined(WIN32)&&!defined(_POSIX_))
        for (i = 0; i < ac; i++)
        {
                spawn_args[i] = malloc ((strlen (av[i])
+ 1) * sizeof (char));
                MALLOC_CHECK (spawn_args[i]);
                strcpy (spawn_args[i], av[i]);
        }
        spawn_args[ac] = NULL;
#endif

        if (verbose >= 0)
                {
                fprintf (stderr,
                        "%s Population Generator
(Version %d.%d.%d%s)\n",
                        NAME, VERSION, RELEASE,
MODIFICATION, PATCH);
                fprintf (stderr, "Copyright %s %s\n",
TPC, C_DATES);
                }
```

```
        load_dists ();
        /* have to do this after init */
        tdefs[NATION].base = nations.count;
        tdefs[REGION].base = regions.count;

        /*
        * updates are never parallelized
        */
        if (updates)
                {
                /*
                 * set RNG to start generating rows
beyond SF=scale
                 */
                set_state (ORDER, scale, 1, 2, (long
*)&i);
                rowcnt = tdefs[ORDER_LINE].base / 10000
* scale * refresh;
                if (step > 0)
                        {
                        /*
                         * adjust RNG for any prior
update generation
                         */
                        sd_order(0, rowcnt * (step -
1));
                        sd_line(0, rowcnt * (step -
1));
                        upd_num = step - 1;
                        }
                else
                        upd_num = 0;

                while (upd_num < updates)
                        {
                        if (verbose > 0)
                                fprintf (stderr,
                                "Generating update pair
#%d for %s [pid: %d]",
                                upd_num + 1,
tdefs[ORDER_LINE].comment, DSS_PROC);
                        insert_orders_segment=0;
                        insert_lineitem_segment=0;
                        delete_segment=0;
                        minrow = upd_num * rowcnt + 1;
                        gen_tbl (ORDER_LINE, minrow,
rowcnt, upd_num + 1);
                        if (verbose > 0)
                                fprintf (stderr,
"done.\n");
                        pr_drange (ORDER_LINE, minrow,
rowcnt, upd_num + 1);
                        upd_num++;
                        }

                exit (0);
                }

        /**
        ** actual data generation section starts here
        **/
/*
 * open database connection or set all the file names,
as appropriate
 */
        if (direct)
                prep_direct ((db_name) ? db_name :
DBNAME);
        else if (fnames)
                for (i = PART; i <= REGION; i++)
                {
                        if (table & (1 << i))
                                if (set_files (i, -1))
                                {
                                        fprintf (stderr,
"Load aborted!\n");
                                        exit (1);
                                }
                }

/*
 * traverse the tables, invoking the appropriate data
generation routine for any to be built
 */
        for (i = PART; i <= REGION; i++)
                if (table & (1 << i))
                {
#ifdef ORACLE
#ifdef DEBUG
                        row_count = 0;
                        for (j=0; j<numpart; j++) {
                          part_row[j] = 0;
                        }
                        sprintf(debug_fn,"%s.%s.%d.out",
"dbgen", tdefs[i].name, step+1);
                        if ((debug_fd = fopen(debug_fn, "w+"))
== NULL) {
                          fprintf(stderr, "Problem opening
%s\n", debug_fn);
                          exit(-1);
                        }
#endif /* DEBUG */
#endif /* ORACLE */
                                if (children > 1 && i < NATION)
                                        if (step >= 0)
                                        {
                                                if (validate)
                                                {
                                                        INTERNAL_
ERROR("Cannot validate parallel data generation");
                                                }
                                                else
                                                        partial
(i, step);
                                        }
#ifdef DOS
                                        else
                                        {
                                                fprintf (stderr,
                                                        "Parallel
load is not supported on your platform.\n");
                                                exit (1);
                                        }
#else
                                        else
                                        {
                                                if (validate)
                                                {
                                                        INTERNAL_
ERROR("Cannot validate parallel data generation");
                                                }
                                                else
                                                        pload
(i);
                                        }
#endif /* DOS */
                                        else
                                        {
                                                minrow = 1;
                                                if (i < NATION)
                                                        rowcnt =
tdefs[i].base * scale;
                                                else
                                                        rowcnt =
tdefs[i].base;
                                                if (verbose > 0)
                                                        fprintf
(stderr, "%s data for %s [pid: %ld]",
                                                        (validate
)?"Validating":"Generating", tdefs[i].comment,
DSS_PROC);
                                                gen_tbl (i,
minrow, rowcnt, upd_num);
                                                if (verbose > 0)
                                                        fprintf
(stderr, "done.\n");
                                        }
                                        if (validate)
                                                printf("Validati
on checksum for %s at %d GB: %0x\n",

tdefs[i].name, scale, tdefs[i].vtotal);
                }

                if (direct)
                        close_direct ();

                return (0);
}
```

==========

part.c

```
/*
==========
 |       Copyright (c) 1997 , 1999 Oracle Corp,
Redwood Shores, CA        |
```

```
 |                   ADVANCED TECH PERFORMANCE GROUP
|
 |                         All Rights Reserved
|
 +=================================================
==============+
 | FILENAME
 |    part.c
 | DESCRIPTION
 |    Functions to support partitioning.
 | MODIFIED
 |    pswong    03/13/97 - created
 +=================================================
==============*/

#include "dss.h"
#include "dsstypes.h"
#include <string.h>
#include "part.h"

extern FILE **out;
extern FILE *par;
extern char delim;
char outname[FLEN];
extern int use_outparam;
FILE *outparam;
extern long numpart;
extern vals *values;

#ifdef DEBUG
extern long row_count;
extern long *part_row;
extern int step;
extern FILE* debug_fd;
#endif /* DEBUG */

void process_param_file()
{

  char ofile[128];
  char line[128];
  char *pos, *str;
  int i,j;
  int retcode;
  struct stat fstats;

  /* get delimiter */

  fscanf(par, "%c\n", &delim);

  /* get number of partitions */

  fscanf(par, "%ld\n", &numpart);

  /* malloc some structures */

  values = (vals *) malloc(sizeof(vals));
  out = (FILE **) malloc(numpart * sizeof(FILE *));

#ifdef DEBUG
  part_row = (long *) malloc(sizeof(long) * numpart);
#endif /* DEBUG */
  /* extract the column types */

  fscanf(par, "%d\n", &(values->type));

  /* extract column position */

  fscanf(par, "%d\n", &(values->pos));

  /* setup output file descriptors */

  for (j=0; j<numpart; j++) {
    if (use_outparam) {
      /* get the file names */
      if (fscanf(outparam, "%s\n", ofile) == EOF) {
        fprintf(stderr, "Error: Number of entries in
output parameter file less than the number of
partitions\n");
        (void) Exitprog(-1);
      }
    } else {
      sprintf(ofile, "%s.%d", (char *) outname,
(j+1));
    }
    out[j] = fopen(ofile, "w");
    if (out[j] == NULL) {
      fprintf(stderr, "Unable to open file '%s'.\n",
ofile);
      fprintf(stderr, "%s: %s\n", ofile,
strerror(errno));
      (void) Exitprog(-1);
    }
  }

  /* allocate memory for the values */

  switch(values->type) {
  case NUM_TYP:
    values->vals = (double *) malloc(sizeof(double) *
numpart);
    break;
  case STR_TYP:
    values->vals = (char **) malloc(sizeof(char *) *
numpart);
    break;
  case DAT_TYP:
    values->vals = (dtyp *) malloc(sizeof(dtyp) *
numpart);
    break;
  }


  /* now read the values              */
  /* could have easily done by fscanf */

  for (j=0; j<numpart; j++) {
    if (fgets(line, 127, par) == NULL) {
      fprintf(stderr, "Error: number of boundaries
definition does not equal to the number of
partitons.\n");
      Exitprog(-1);
    }
    str = strtok(line, DELIM);
    switch(values->type) {
    case NUM_TYP:
      ((double *)values->vals)[j] = (double)
atof(str);
      break;
    case STR_TYP:
      ((char **)values->vals)[j] = (char *)
malloc(strlen(str) + 1);
      strcpy(((char **)values->vals)[j], str);
      break;
    case DAT_TYP:
      ASS_DATE(((dtyp *)values->vals)[j],str);
      break;
    }
  }
}


void Exitprog(errn)
     int errn;
{
  int i,j;

  /* free some memory */

  if (values->type == STR_TYP) {
    for(i=0;i<numpart;i++)
      free(((char **)values->vals)[i]);
  }

  for (i=0;i<numpart;i++)
    fclose(out[i]);

#ifdef DEBUG
  if (debug_fd != NULL) {
    fprintf(debug_fd, "dbgen was unsuccessful for step
%d\n", step);
    fprintf(debug_fd, "total number of rows generated:
%ld\n", (long) row_count);
    for(j=0;j<numpart;j++) {
      fprintf(debug_fd, "partition: %d rows: %ld\n",
              j+1, part_row[j]);
    }
    free(part_row);
    fclose(debug_fd);
  }
#endif /* DEBUG */

  free(values->vals);
  free(out);
  free(values);
  exit(errn);
}
```

```
int find_partition(cval,type)
     void *cval;
     int type;
{
  char tmpdate[16];
  dtyp vdat;

  if (type != values->type)
  {
      fprintf(stderr,"Type mismatch between parameter
file and the column type\n");
      Exitprog(-1);
  }
  switch(values->type) {
  case NUM_TYP:
    return (part_search((double *)values->vals,
(double *) cval, NUM_TYP,
                        fltcmp));
  case STR_TYP:
    return (part_search((char **)values->vals, (char
*) cval, STR_TYP,
                        strgcmp));
  case DAT_TYP:
    strcpy(tmpdate, (char*)cval);
    ASS_DATE(vdat,(char *)tmpdate);
    return (part_search((dtyp *)values->vals, (dtyp *)
&vdat, DAT_TYP,
                        datcmp));
  }
}


int fltcmp(a,b)
     void *a;
     void *b;
{
  if (*((double *)a) == *((double *)b))
    return 0;
  else {
    if (*((double *)a) > *((double *)b)) return 1;
    else return -1;
  }
}



int strgcmp(a,b)
     void *a;
     void *b;
{
  int ret;

  ret = strncmp((char *)a, (char *)b,
                MINV(strlen((char *)a),strlen((char
*)b)));
  return(ret);
}


int datcmp(a,b)
     void *a;
     void *b;
{
  if ((((dtyp *)a)->year) > (((dtyp *)b)->year)) {
    return 1;
  }  else {
    if ((((dtyp *)a)->year) < (((dtyp *)b)->year)) {
      return -1;
    } else {
      if ((((dtyp *)a)->month) > (((dtyp *)b)->month))
{
        return 1;
      } else {
        if ((((dtyp *)a)->month) < (((dtyp
*)b)->month)) {
          return -1;
        } else {
          if ((((dtyp *)a)->day) > (((dtyp *)b)->day))
{
            return 1;
          } else {
            if ((((dtyp *)a)->day) < (((dtyp
*)b)->day)) {
              return -1;
            } else {
              return 0;
            }
          }
        }
      }
    }
  }
}


void *ptr_arith(ptr, inc, type)
     void *ptr;
     int inc;
     int type;
{
  switch(type) {
  case NUM_TYP:
    return ((double *)ptr + inc);
    break;
  case STR_TYP:
    return *((char **)ptr + inc);
    break;
  case DAT_TYP:
    return ((dtyp *)ptr + inc);
  }
}


/* Simple search, returns index into the array     */
/* arr is the array of size numpart to be searched */
/* item is the item to search for                  */
/* fp is the pointer to the comparison function    */
/* fp(&a,&b) should return 0 if a == b             */
/*                         1 if a >  b             */
/*                        -1 if a <  b             */

int part_search(arr,item,type,fp)
     void *arr;
     void *item;
     int type;
     int (* fp)();
{

  static int hint = 0;       /* hint */
  int i;

  while (1) {
    /* if we are smaller than the very first partition
*/
    if (hint == 0) {
      if (fp(ptr_arith(arr,0,type),item) > 0)
        return 0;
      else {
        hint++;
        continue;
      }
    }

    /* if we are larger than the very last partition
*/
    if (hint == (numpart - 1)) {
      if (fp(ptr_arith(arr,hint,type),item) <= 0)
        return (numpart-1);
      else {
        if (fp(ptr_arith(arr,(hint-1),type),item) <=
0)
          return hint;
        else {
          hint--;
          continue;
        }
      }
    }

    if (fp(ptr_arith(arr,hint,type),item) > 0) {
      /* let's see if item is in the current partition
*/
      if (fp(ptr_arith(arr,(hint-1),type),item) <= 0)
        return hint;
      else
        hint--;
    } else hint++;
  }
}

int find_partition_cust(customer_t *c)
{
double tmpnum;
    switch (values->pos)
    {
        case 1:
            tmpnum = c->custkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
```

```
            break;
        case 2:
            return(find_partition(c->mktsegment,
STR_TYP));
            break;
        case 3:
            tmpnum = c->nation_code;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 4:
            return(find_partition(c->name, STR_TYP));
            break;
        case 5:
            return(find_partition(c->address,
STR_TYP));
            break;
        case 6:
            return(find_partition(c->phone, STR_TYP));
            break;
        case 7:
            tmpnum = c->acctbal;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 8:
            return(find_partition(c->comment,
STR_TYP));
            break;
        default:
            fprintf(stderr,"Incorrect column number
for partitioning.\n");
            Exitprog(-1);
            break;
    }
}

int find_partition_order(order_t *o)
{
double tmpnum;
    switch (values->pos)
    {
        case 1:
            return(find_partition(o->odate, DAT_TYP));
            break;
        case 2:
            tmpnum = *(o->okey);
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 3:
            tmpnum = o->custkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 4:
            return(find_partition(o->opriority,
STR_TYP));
            break;
        case 5:
            tmpnum = o->spriority;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 6:
            return(find_partition(o->clerk, STR_TYP));
            break;
        case 7:
            return(find_partition(o->orderstatus,
STR_TYP));
            break;
        case 8:
            tmpnum = o->totalprice;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 9:
            return(find_partition(o->comment,
STR_TYP));
            break;
        default:
            fprintf(stderr,"Incorrect column number
for partitioning.\n");
            Exitprog(-1);
            break;
    }
}

int find_partition_line(order_t *o, int i)
{
double tmpnum;

    switch (values->pos)
    {
        case 1:
            return(find_partition(o->l[i].sdate,
DAT_TYP));
            break;
        case 2:
            tmpnum = *(o->okey);
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 3:
            tmpnum = o->l[i].discount;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 4:
            tmpnum = o->l[i].eprice;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 5:
            tmpnum = o->l[i].suppkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 6:
            tmpnum = o->l[i].quantity;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 7:
            return(find_partition(o->l[i].rflag[0],
STR_TYP));
            break;
        case 8:
            tmpnum = o->l[i].partkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 9:
            return(find_partition(o->l[i].lstatus[0],
STR_TYP));
            break;
        case 10:
            tmpnum = o->l[i].tax;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 11:
            return(find_partition(o->l[i].cdate,
DAT_TYP));
            break;
        case 12:
            return(find_partition(o->l[i].rdate,
DAT_TYP));
            break;
        case 13:
            return(find_partition(o->l[i].shipmode,
STR_TYP));
            break;
        case 14:
            tmpnum = o->l[i].lcnt;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 15:

return(find_partition(o->l[i].shipinstruct, STR_TYP));
            break;
        case 16:
            return(find_partition(o->l[i].comment,
STR_TYP));
            break;
        default:
            fprintf(stderr,"Incorrect column number
for partitioning.\n");
            Exitprog(-1);
            break;
    }
}

int find_partition_part(part_t *p)
{
double tmpnum;

    switch (values->pos)
    {
```

```
        case 1:
            tmpnum = p->partkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 2:
            return(find_partition(p->type, STR_TYP));
            break;
        case 3:
            tmpnum = p->size;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 4:
            return(find_partition(p->brand, STR_TYP));
            break;
        case 5:
            return(find_partition(p->name, STR_TYP));
            break;
        case 6:
            return(find_partition(p->container,
STR_TYP));
            break;
        case 7:
            return(find_partition(p->mfgr, STR_TYP));
            break;
        case 8:
            tmpnum = p->retailprice;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 9:
            return(find_partition(p->comment,
STR_TYP));
            break;
        default:
            fprintf(stderr,"Incorrect column number
for partitioning.\n");
            Exitprog(-1);
            break;
    }
}

int find_partition_psupp(partsupp_t *ps)
{
double tmpnum;

    switch (values->pos)
    {
        case 1:
            tmpnum = ps->partkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 2:
            tmpnum = ps->suppkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 3:
            tmpnum = ps->qty;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 4:
            tmpnum = ps->scost;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 5:
            return(find_partition(ps->comment,
STR_TYP));
            break;
        default:
            fprintf(stderr,"Incorrect column number
for partitioning.\n");
            Exitprog(-1);
            break;
    }
}

int find_partition_supp(supplier_t *s)
{
double tmpnum;

    switch (values->pos)
    {
        case 1:
            tmpnum = s->suppkey;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 2:
            tmpnum = s->nation_code;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        case 3:
            return(find_partition(s->comment,
STR_TYP));
            break;
        case 4:
            return(find_partition(s->name, STR_TYP));
            break;
        case 5:
            return(find_partition(s->address,
STR_TYP));
            break;
        case 6:
            return(find_partition(s->phone, STR_TYP));
            break;
        case 7:
            tmpnum = s->acctbal;
            return(find_partition(&(tmpnum),
NUM_TYP));
            break;
        default:
            fprintf(stderr,"Incorrect column number
for partitioning.\n");
            Exitprog(-1);
            break;
    }
}
```

==========

part.h

==========

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>

#define FLEN 128

typedef struct dt {
  int year;
  int month;
  int day;
} dtyp;

typedef struct val {
  int type;
  int pos;
  void *vals;
} vals;

#define ASS_DATE(dat,str) \
{ dat.year = atoi(strtok(str,"-\n")); \
  dat.month = atoi(strtok(NULL,"-\n")); \
  dat.day = atoi(strtok(NULL,"-\n"));}

#define MINV(x,y) ((x < y) ? x : y)

/* some marco to do the print */

#ifndef SEPARATOR
#define SEPARATOR '|' /* field spearator for generated
flat files */
#endif

#define  PPR_STR(f,str,len) \
        if (columnar) sprintf(f, "%-*s", len, str); \
        else sprintf(f, "%s%c", str, SEPARATOR); \
        f += strlen(f)

#define  PPR_VSTR(f,str,len) PPR_STR(f,str,len)

#define  PPR_INT(f,long)    \
        if (columnar) sprintf(f, "%12ld", long); \
        else sprintf(f,"%ld%c", long, SEPARATOR); \
        f += strlen(f)

#define  PPR_BCD2(f,high,low)    \
        if (high == 0) \
           if (columnar) sprintf(f, "%12ld", low); \
           else sprintf(f,"%ld%c", low, SEPARATOR); \
        else    \
```

```
        if (columnar) sprintf(f, "%5ld%07ld", high,
low); \
        else sprintf(f,"%ld%07ld%c", high, low,
SEPARATOR); \
        f += strlen(f)

#define  PPR_KEY(f,long)    \
        sprintf(f,"%ld", long); \
        f += strlen(f)

#define MONEY_COL_FMT "%12ld.%02ld"
#define PPR_MONEY(fp,flt)    \
        if (columnar) sprintf(fp, MONEY_COL_FMT, \
                flt/100, ((flt < 0)?-flt:flt)%100); \
        else sprintf(fp,"%ld.%02ld%c", \
                flt/100, ((flt < 0)?-flt:flt)%100,
SEPARATOR); \
        fp += strlen(fp)

#define  PPR_CHR(fp,chr)   \
        if (columnar) sprintf(fp, "%c ", chr); \
        else sprintf(fp,"%c%c", chr, SEPARATOR); \
        fp += strlen(fp)

#define  PPR_STRT(dest,ptr)  (ptr = dest)
#define  PPR_END(fp)    sprintf(fp, "\n")   /* finish
the record here */

#ifdef MDY_DATE
#define  PR_DATE(tgt, yr, mn, dy)       \
   sprintf(tgt, "%02d-%02d-19%02d", mn, dy, yr)
#else
#define  PR_DATE(tgt, yr, mn, dy)       \
   sprintf(tgt, "19%02d-%02d-%02d", yr, mn, dy)
#endif /* DATE_FORMAT */

/* Could add a integer type and a character type to
allow faster processing. */
#define NUM_TYP 0       /* number type    */
#define STR_TYP 1       /* str/char type  */
#define DAT_TYP 2       /* date type      */

#define DELIM "|\n"

void Exitprog();
void process_param_file();
int fltcmp();
int datcmp();
int strgcmp();
int part_search();
int find_partition();
```

==========

print.c

==========

```
/* @(#)print.c 2.1.8.2 */
/* generate flat files for data load */
#include <stdio.h>
#ifndef VMS
#include <sys/types.h>
#endif
#if defined(SUN)
#include <unistd.h>
#endif
#include <math.h>

#include "dss.h"
#include "dsstypes.h"
#include <string.h>

#ifdef ORACLE
#include "part.h"

extern vals *values;
extern FILE **out;              /* output fd array
*/
extern long partnum;           /* partition number
*/
extern int partition;
char tmpdate[16];              /* temporary storage to
store dates */

int slen;                      /* current string
length              */
int wlen;                      /* number of bytes
written           */

#ifdef DEBUG
long row_count;                /* total number of rows
generated     */
long *part_row;                /* number of rows per
partition       */
#endif /* DEBUG */

extern void Exitprog();

#endif /* ORACLE */

/*
 * Function Prototypes
 */
FILE *print_prep PROTO((int table, int update));
int pr_drange PROTO((int tbl, long min, long cnt, long
num));


FILE *
print_prep(int table, int update)
{
        char upath[128];
        FILE *res;

        if (updates)
                {
                if (update > 0) /* updates */
                        if ( insert_segments )
                                {
                                int this_segment;
                                if(strcmp(tdefs[table].n
ame,"orders.tbl"))
                                        this_segment=++i
nsert_orders_segment;
                                else
                                        this_segment=++i
nsert_lineitem_segment;
                                sprintf(upath,
"%s%c%s.u%d.%d",
                                        env_config(PATH_
TAG, PATH_DFLT),
                                        PATH_SEP,
tdefs[table].name, update%10000,this_segment);
                                }
                        else
                                {
                                sprintf(upath,
"%s%c%s.u%d",
                                env_config(PATH_TAG,
PATH_DFLT),
                                PATH_SEP,
tdefs[table].name, update);
                                }
                else /* deletes */
                        if ( delete_segments )
                                {
                                ++delete_segment;
                                sprintf(upath,
"%s%cdelete.u%d.%d",
                                        env_config(PATH_
TAG, PATH_DFLT), PATH_SEP, -update%10000,
                                        delete_segment);
                                }
                        else
                                {
                                sprintf(upath,
"%s%cdelete.%d",
                                env_config(PATH_TAG,
PATH_DFLT), PATH_SEP, -update);
                                }
                return(fopen(upath, "w"));
                }
    res = tbl_open(table, "w");
    OPEN_CHECK(res, tdefs[table].name);
    return(res);
}

int
dbg_print(int format, FILE *tgt, void *data, int len,
int sep)
{
        int dollars,
                cents;

        switch(format)
        {
        case DT_STR:
                if (columnar)
                        fprintf(tgt, "%-*s", len, (char
*)data);
```

```
                else
                        fprintf(tgt, "%s", (char
*)data);
                break;
#ifdef MVS
        case DT_VSTR:
                /* note: only used in MVS, assumes
columnar output */
                fprintf(tgt, "%c%c%-*s",
                        (len >> 8) & 0xFF, len & 0xFF,
len, (char *)data);
                break;
#endif /* MVS */
        case DT_INT:
                if (columnar)
                        fprintf(tgt, "%12ld",
(long)data);
                else
                        fprintf(tgt, "%ld",
(long)data);
                break;
        case DT_HUGE:
#ifndef SUPPORT_64BITS
         if (*(long *)((long *)data + 1) == 0) \
            if (columnar) fprintf(tgt, "%12ld", *(long
*)data);
            else fprintf(tgt, "%ld", *(long *)data);
         else
            if (columnar) fprintf(tgt, "%5ld%07ld",
                                *(long *)((long *)data +
1), *(long *)data);
            else fprintf(tgt,"%ld%07ld",
                                *(long *)((long *)data +
1), *(long *)data);
#else
        /* printf ("before writing to disk: data is
%lld\n",*(DSS_HUGE *)data);*/

                fprintf(tgt, HUGE_FORMAT, *(DSS_HUGE
*)data);
#endif /* SUPPORT_64BITS */
                break;
        case DT_KEY:
                fprintf(tgt, "%ld", (long)data);
                break;
        case DT_MONEY:
                cents = (long)data;
                if (cents < 0)
                        {
                        fprintf(tgt, "-");
                        cents = -cents;
                        }
                dollars = cents / 100;
                cents %= 100;
                if (columnar)
                        fprintf(tgt, "%12ld.%02ld",
dollars, cents);
                else
                        fprintf(tgt, "%ld.%02ld",
dollars, cents);
                break;
        case DT_CHR:
                if (columnar)
#ifdef LINUX
                        fprintf(tgt, "%c ", *(char
*)data);
#else
                        fprintf(tgt, "%c ",
(char)data);
#endif
                else
#ifdef LINUX
                        fprintf(tgt, "%c", *(char
*)data);
#else
                        fprintf(tgt, "%c", (char)data);
#endif
                break;
        }

#ifdef EOL_HANDLING
        if (sep)
#endif /* EOL_HANDLING */
        if (!columnar && (sep != -1))
                fprintf(tgt, "%c", SEPARATOR);

        return(0);
}

int
pr_cust(customer_t *c, int mode)
{
static FILE *fp = NULL;

#ifdef ORACLE
    if (partition) {
       partnum = find_partition_cust(c);
       fp = out[partnum];
    } else if (fp == NULL)
        fp = print_prep(CUST, 0);

   PR_STRT(fp);
   PR_INT(fp, c->custkey);
   PR_STR(fp, c->mktsegment, C_MSEG_LEN);
   PR_INT(fp, c->nation_code);
   PR_VSTR(fp, c->name, C_NAME_LEN);
   PR_VSTR(fp, c->address,
        (columnar)?(long)(ceil(C_ADDR_LEN *
V_STR_HGH)):c->alen);
   PR_STR(fp, c->phone, PHONE_LEN);
   PR_MONEY(fp, c->acctbal);
   PR_VSTR_LAST(fp, c->comment,
        (columnar)?(long)(ceil(C_CMNT_LEN *
V_STR_HGH)):c->clen);
   PR_END(fp);
#else /* !ORACLE */
   if (fp == NULL)
        fp = print_prep(CUST, 0);

   PR_STRT(fp);
   PR_INT(fp, c->custkey);
   PR_VSTR(fp, c->name, C_NAME_LEN);
   PR_VSTR(fp, c->address,
        (columnar)?(long)(ceil(C_ADDR_LEN *
V_STR_HGH)):c->alen);
   PR_INT(fp, c->nation_code);
   PR_STR(fp, c->phone, PHONE_LEN);
   PR_MONEY(fp, c->acctbal);
   PR_STR(fp, c->mktsegment, C_MSEG_LEN);
   PR_VSTR_LAST(fp, c->comment,
        (columnar)?(long)(ceil(C_CMNT_LEN *
V_STR_HGH)):c->clen);
   PR_END(fp);
#endif /* !ORACLE */

   return(0);
}

/*
 * print the numbered order
 */
int
pr_order(order_t *o, int mode)
{
    static FILE *fp_o = NULL;
    static int last_mode = 0;

#ifdef ORACLE
    if (partition)
    {
        partnum = find_partition_order(o);
        fp_o = out[partnum];
        if (mode != last_mode)
            last_mode = mode;
    } else
    {
        if (fp_o == NULL || mode != last_mode)
        {
        if (fp_o)
            fclose(fp_o);
        fp_o = print_prep(ORDER, mode);
        last_mode = mode;
        }
    }
    PR_STRT(fp_o);
    PR_STR(fp_o, o->odate, DATE_LEN);
    /* printf ("o->okey=%lld\n",*o->okey); */
    PR_HUGE(fp_o, o->okey);
    PR_INT(fp_o, o->custkey);
    PR_STR(fp_o, o->opriority, O_OPRIO_LEN);
    PR_INT(fp_o, o->spriority);
    PR_STR(fp_o, o->clerk, O_CLRK_LEN);
#ifdef LINUX
    PR_CHR(fp_o, &(o->orderstatus));
#else
    PR_CHR(fp_o, o->orderstatus);
#endif
    PR_MONEY(fp_o, o->totalprice);
    PR_VSTR_LAST(fp_o, o->comment,
       (columnar)?(long)(ceil(O_CMNT_LEN *
```

```
V_STR_HGH)):o->clen);
    PR_END(fp_o);
#else /* !ORACLE */
    if (fp_o == NULL || mode != last_mode)
        {
        if (fp_o)
            fclose(fp_o);
        fp_o = print_prep(ORDER, mode);
        last_mode = mode;
        }
    PR_STRT(fp_o);
    PR_HUGE(fp_o, o->okey);
    PR_INT(fp_o, o->custkey);
    PR_CHR(fp_o, o->orderstatus);
    PR_MONEY(fp_o, o->totalprice);
    PR_STR(fp_o, o->odate, DATE_LEN);
    PR_STR(fp_o, o->opriority, O_OPRIO_LEN);
    PR_STR(fp_o, o->clerk, O_CLRK_LEN);
    PR_INT(fp_o, o->spriority);
      (columnar)?(long)(ceil(O_CMNT_LEN *
V_STR_HGH)):o->clen);
    PR_END(fp_o);
#endif /* !ORACLE */

    return(0);
}

/*
 * print an order's lineitems
 */
int
pr_line(order_t *o, int mode)
{
    static FILE *fp_l = NULL;
    static int last_mode = 0;
    long      i;

#ifdef ORACLE
    if (!partition)
    {
        if (fp_l == NULL || mode != last_mode)
        {
            if (fp_l)
                fclose(fp_l);
            fp_l = print_prep(LINE, mode);
            last_mode = mode;
        }
    } else {
        if ( mode != last_mode)
            last_mode = mode;
    }

    for (i = 0; i < o->lines; i++)
    {
        if (partition) {
            partnum = find_partition_line(o, i);
            fp_l = out[partnum];
        }
        PR_STRT(fp_l);
        PR_STR(fp_l, o->l[i].sdate, DATE_LEN);
        PR_HUGE(fp_l, o->okey);
        PR_MONEY(fp_l, o->l[i].discount);
        PR_MONEY(fp_l, o->l[i].eprice);
        PR_INT(fp_l, o->l[i].suppkey);
        PR_INT(fp_l, o->l[i].quantity);
#ifdef LINUX
        PR_CHR(fp_l, &(o->l[i].rflag[0]));
#else
        PR_CHR(fp_l, o->l[i].rflag[0]);
#endif
        PR_INT(fp_l, o->l[i].partkey);
#ifdef LINUX
        PR_CHR(fp_l, &(o->l[i].lstatus[0]));
#else
        PR_CHR(fp_l, o->l[i].lstatus[0]);
#endif
        PR_MONEY(fp_l, o->l[i].tax);
        PR_STR(fp_l, o->l[i].cdate, DATE_LEN);
        PR_STR(fp_l, o->l[i].rdate, DATE_LEN);
        PR_STR(fp_l, o->l[i].shipmode, L_SMODE_LEN);
        PR_INT(fp_l, o->l[i].lcnt);
        PR_STR(fp_l, o->l[i].shipinstruct,
L_INST_LEN);
        PR_VSTR_LAST(fp_l, o->l[i].comment,
           (columnar)?(long)(ceil(L_CMNT_LEN *
V_STR_HGH)):o->l[i].clen);
        PR_END(fp_l);
    }
#else /* !ORACLE */
    if (fp_l == NULL || mode != last_mode)
        {
        if (fp_l)
            fclose(fp_l);
        fp_l = print_prep(LINE, mode);
        last_mode = mode;
        }

    for (i = 0; i < o->lines; i++)
        {
        PR_STRT(fp_l);
        PR_HUGE(fp_l, o->l[i].okey);
        PR_INT(fp_l, o->l[i].partkey);
        PR_INT(fp_l, o->l[i].suppkey);
        PR_INT(fp_l, o->l[i].lcnt);
        PR_INT(fp_l, o->l[i].quantity);
        PR_MONEY(fp_l, o->l[i].eprice);
        PR_MONEY(fp_l, o->l[i].discount);
        PR_MONEY(fp_l, o->l[i].tax);
        PR_CHR(fp_l, o->l[i].rflag[0]);
        PR_CHR(fp_l, o->l[i].lstatus[0]);
        PR_STR(fp_l, o->l[i].sdate, DATE_LEN);
        PR_STR(fp_l, o->l[i].cdate, DATE_LEN);
        PR_STR(fp_l, o->l[i].rdate, DATE_LEN);
        PR_STR(fp_l, o->l[i].shipinstruct,
L_INST_LEN);
        PR_STR(fp_l, o->l[i].shipmode, L_SMODE_LEN);
        PR_VSTR_LAST(fp_l, o->l[i].comment,
           (columnar)?(long)(ceil(L_CMNT_LEN *
V_STR_HGH)):o->l[i].clen);
        PR_END(fp_l);
        }
#endif /* !ORACLE */

   return(0);
}

/*
 * print the numbered order *and* its associated
lineitems
 */
int
pr_order_line(order_t *o, int mode)
{
    tdefs[ORDER].name = tdefs[ORDER_LINE].name;
    pr_order(o, mode);
    pr_line(o, mode);

    return(0);
}

/*
 * print the given part
 */
int
pr_part(part_t *part, int mode)
{
static FILE *p_fp = NULL;

#ifdef ORACLE

   if (partition)
   {
       partnum = find_partition_part(part);
       p_fp = out[partnum];
   } else
   {
   if (p_fp == NULL)
       p_fp = print_prep(PART, 0);
   }

   PR_STRT(p_fp);
   PR_INT(p_fp, part->partkey);
   PR_VSTR(p_fp, part->type,
       (columnar)?(long)P_TYPE_LEN:part->tlen);
   PR_INT(p_fp, part->size);
   PR_STR(p_fp, part->brand, P_BRND_LEN);
   PR_VSTR(p_fp, part->name,
       (columnar)?(long)P_NAME_LEN:part->nlen);
   PR_STR(p_fp, part->container, P_CNTR_LEN);
   PR_STR(p_fp, part->mfgr, P_MFG_LEN);
   PR_MONEY(p_fp, part->retailprice);
   PR_VSTR_LAST(p_fp, part->comment,
       (columnar)?(long)(ceil(P_CMNT_LEN *
V_STR_HGH)):part->clen);
   PR_END(p_fp);
#else /* !ORACLE */
    if (p_fp == NULL)
        p_fp = print_prep(PART, 0);

   PR_STRT(p_fp);
```

```
   PR_INT(p_fp, part->partkey);
   PR_VSTR(p_fp, part->name,
       (columnar)?(long)P_NAME_LEN:part->nlen);
   PR_STR(p_fp, part->mfgr, P_MFG_LEN);
   PR_STR(p_fp, part->brand, P_BRND_LEN);
   PR_VSTR(p_fp, part->type,
       (columnar)?(long)P_TYPE_LEN:part->tlen);
   PR_INT(p_fp, part->size);
   PR_STR(p_fp, part->container, P_CNTR_LEN);
   PR_MONEY(p_fp, part->retailprice);
   PR_VSTR_LAST(p_fp, part->comment,
       (columnar)?(long)(ceil(P_CMNT_LEN *
V_STR_HGH)):part->clen);
   PR_END(p_fp);
#endif /* !ORACLE */

   return(0);
}

/*
 * print the given part's suppliers
 */
int
pr_psupp(part_t *part, int mode)
{
    static FILE *ps_fp = NULL;
    long      i;

#ifdef ORACLE
    if ((!partition) && (ps_fp == NULL))
        ps_fp = print_prep(PSUPP, mode);
#else /* !ORACLE */
    if (ps_fp == NULL)
        ps_fp = print_prep(PSUPP, mode);
#endif /* !ORACLE */

   for (i = 0; i < SUPP_PER_PART; i++)
      {
#ifdef ORACLE
      if (partition)
      {
          partnum = find_partition_psupp(part);
          ps_fp = out[partnum];
      }
#endif /* ORACLE */
      PR_STRT(ps_fp);
      PR_INT(ps_fp, part->s[i].partkey);
      PR_INT(ps_fp, part->s[i].suppkey);
#ifdef ORACLE
      PR_MONEY(ps_fp, part->s[i].scost);
      PR_INT(ps_fp, part->s[i].qty);
#endif /* ORACLE */
#ifndef ORACLE
      PR_INT(ps_fp, part->s[i].qty);
      PR_MONEY(ps_fp, part->s[i].scost);
#endif /* !ORACLE */
      PR_VSTR_LAST(ps_fp, part->s[i].comment,
       (columnar)?(long)(ceil(PS_CMNT_LEN *
V_STR_HGH)):part->s[i].clen);
      PR_END(ps_fp);
      }

   return(0);
}

/*
 * print the given part *and* its suppliers
 */
int
pr_part_psupp(part_t *part, int mode)
{
    tdefs[PART].name = tdefs[PART_PSUPP].name;
    pr_part(part, mode);
    pr_psupp(part, mode);

    return(0);
}

int
pr_supp(supplier_t *supp, int mode)
{
static FILE *fp = NULL;

#ifdef ORACLE
   if (partition)
   {
       partnum = find_partition_supp(supp);
       fp = out[partnum];
   } else if (fp == NULL)
        fp = print_prep(SUPP, mode);
   PR_STRT(fp);
   PR_INT(fp, supp->suppkey);
   PR_INT(fp, supp->nation_code);
   PR_VSTR(fp, supp->comment,
       (columnar)?(long)(ceil(S_CMNT_LEN *
V_STR_HGH)):supp->clen);
   PR_STR(fp, supp->name, S_NAME_LEN);
   PR_VSTR(fp, supp->address,
       (columnar)?(long)(ceil(S_ADDR_LEN *
V_STR_HGH)):supp->alen);
   PR_STR(fp, supp->phone, PHONE_LEN);
   PR_MONEY(fp, supp->acctbal);
   PR_END(fp);
#else /* !ORACLE */
   if (fp == NULL)
        fp = print_prep(SUPP, mode);
   PR_STRT(fp);
   PR_INT(fp, supp->suppkey);
   PR_STR(fp, supp->name, S_NAME_LEN);
   PR_VSTR(fp, supp->address,
       (columnar)?(long)(ceil(S_ADDR_LEN *
V_STR_HGH)):supp->alen);
   PR_INT(fp, supp->nation_code);
   PR_STR(fp, supp->phone, PHONE_LEN);
   PR_MONEY(fp, supp->acctbal);
   PR_VSTR_LAST(fp, supp->comment,
       (columnar)?(long)(ceil(S_CMNT_LEN *
V_STR_HGH)):supp->clen);
   PR_END(fp);
#endif /* !ORACLE */

   return(0);
}

int
pr_nation(code_t *c, int mode)
{
static FILE *fp = NULL;

   if (fp == NULL)
        fp = print_prep(NATION, mode);

   PR_STRT(fp);
   PR_INT(fp, c->code);
   PR_STR(fp, c->text, NATION_LEN);
   PR_INT(fp, c->join);
   PR_VSTR_LAST(fp, c->comment,
       (columnar)?(long)(ceil(N_CMNT_LEN *
V_STR_HGH)):c->clen);
   PR_END(fp);

   return(0);
}

int
pr_region(code_t *c, int mode)
{
static FILE *fp = NULL;

   if (fp == NULL)
        fp = print_prep(REGION, mode);

   PR_STRT(fp);
   PR_INT(fp, c->code);
   PR_STR(fp, c->text, REGION_LEN);
   PR_VSTR_LAST(fp, c->comment,
       (columnar)?(long)(ceil(R_CMNT_LEN *
V_STR_HGH)):c->clen);
   PR_END(fp);

   return(0);
}

/*
 * NOTE: this routine does NOT use the BCD2_*
routines. As a result,
 * it WILL fail if the keys being deleted exceed 32
bits. Since this
 * would require ~660 update iterations, this seems an
acceptable
 * oversight
 */
int
pr_drange(int tbl, long min, long cnt, long num)
{
    static int  last_num = 0;
    static FILE *dfp = NULL;
    int child = -1;
    long start, last, new;
```

```
        static int rows_per_segment=0;
        static int rows_this_segment=0;

    if (last_num != num)
        {
        if (dfp)
            fclose(dfp);
        dfp = print_prep(tbl, -num);
        if (dfp == NULL)
            return(-1);
        last_num = num;
              rows_this_segment=0;
        }

    start = MK_SPARSE(min, num/ (10000 / refresh));
    last = start - 1;
    for (child=min; cnt > 0; child++, cnt--)
        {
        new = MK_SPARSE(child, num/ (10000 /
refresh));
        if (gen_rng == 1 && new - last == 1)
            {
            last = new;
            continue;
            }
        if (gen_sql)
            {
            fprintf(dfp,
               "delete from %s where %s between %ld
and %ld;\n",
                    tdefs[ORDER].name, "o_orderkey",
start, last);
            fprintf(dfp,
               "delete from %s where %s between %ld
and %ld;\n",
                    tdefs[LINE].name, "l_orderkey",
start, last);
            fprintf(dfp, "commit work;\n");
            }
        else
            if (gen_rng)
                {
                PR_STRT(dfp);
                PR_INT(dfp, start);
                PR_INT(dfp, last);
                PR_END(dfp);
                }
            else
                {
                            if (delete_segments)
                                    {

                                    if(rows_per_segm
ent==0)
                                            rows_per_
segment = (cnt / delete_segments) + 1;
                                    if((++rows_this_
segment) > rows_per_segment)
                                            {
                                            fclose(df
p);
                                            dfp =
print_prep(tbl, -num);
                                            if (dfp
== NULL) return(-1);
                                            last_num
= num;
                                            rows_this
_segment=1;
                                            }
                                    }
                PR_STRT(dfp);
                PR_KEY(dfp, new);
                PR_END(dfp);
                }
        start = new;
        last = new;
        }
    if (gen_rng)
        {
        PR_STRT(dfp);
        PR_INT(dfp, start);
        PR_INT(dfp, last);
        PR_END(dfp);
        }

    return(0);
}

/*
 * verify functions: routines which replace the
pr_routines and generate a pseudo checksum
 * instead of generating the actual contents of the
tables. Meant to allow large scale data
 * validation without requiring a large amount of
storage
 */
int
vrf_cust(customer_t *c, int mode)
{
   VRF_STRT(CUST);
   VRF_INT(CUST, c->custkey);
   VRF_STR(CUST, c->name);
   VRF_STR(CUST, c->address);
   VRF_INT(CUST, c->nation_code);
   VRF_STR(CUST, c->phone);
   VRF_MONEY(CUST, c->acctbal);
   VRF_STR(CUST, c->mktsegment);
   VRF_STR(CUST, c->comment);
   VRF_END(CUST);

   return(0);
}

/*
 * print the numbered order
 */
int
vrf_order(order_t *o, int mode)
{
    VRF_STRT(ORDER);
    VRF_HUGE(ORDER, o->okey);
    VRF_INT(ORDER, o->custkey);
    VRF_CHR(ORDER, o->orderstatus);
    VRF_MONEY(ORDER, o->totalprice);
    VRF_STR(ORDER, o->odate);
    VRF_STR(ORDER, o->opriority);
    VRF_STR(ORDER, o->clerk);
    VRF_INT(ORDER, o->spriority);
    VRF_STR(ORDER, o->comment);
    VRF_END(ORDER);

    return(0);
}

/*
 * print an order's lineitems
 */
int
vrf_line(order_t *o, int mode)
{
       int i;

    for (i = 0; i < o->lines; i++)
        {
        VRF_STRT(LINE);
        VRF_HUGE(LINE, o->l[i].okey);
        VRF_INT(LINE, o->l[i].partkey);
        VRF_INT(LINE, o->l[i].suppkey);
        VRF_INT(LINE, o->l[i].lcnt);
        VRF_INT(LINE, o->l[i].quantity);
        VRF_MONEY(LINE, o->l[i].eprice);
        VRF_MONEY(LINE, o->l[i].discount);
        VRF_MONEY(LINE, o->l[i].tax);
        VRF_CHR(LINE, o->l[i].rflag[0]);
        VRF_CHR(LINE, o->l[i].lstatus[0]);
        VRF_STR(LINE, o->l[i].sdate);
        VRF_STR(LINE, o->l[i].cdate);
        VRF_STR(LINE, o->l[i].rdate);
        VRF_STR(LINE, o->l[i].shipinstruct);
        VRF_STR(LINE, o->l[i].shipmode);
        VRF_STR(LINE, o->l[i].comment);
        VRF_END(LINE);
        }

   return(0);
}

/*
 * print the numbered order *and* its associated
lineitems
 */
int
vrf_order_line(order_t *o, int mode)
{
    vrf_order(o, mode);
    vrf_line(o, mode);

    return(0);
}
```

```
/*
 * print the given part
 */
int
vrf_part(part_t *part, int mode)
{

   VRF_STRT(PART);
   VRF_INT(PART, part->partkey);
   VRF_STR(PART, part->name);
   VRF_STR(PART, part->mfgr);
   VRF_STR(PART, part->brand);
   VRF_STR(PART, part->type);
   VRF_INT(PART, part->size);
   VRF_STR(PART, part->container);
   VRF_MONEY(PART, part->retailprice);
   VRF_STR(PART, part->comment);
   VRF_END(PART);

   return(0);
}

/*
 * print the given part's suppliers
 */
int
vrf_psupp(part_t *part, int mode)
{
    long      i;

   for (i = 0; i < SUPP_PER_PART; i++)
      {
      VRF_STRT(PSUPP);
      VRF_INT(PSUPP, part->s[i].partkey);
      VRF_INT(PSUPP, part->s[i].suppkey);
      VRF_INT(PSUPP, part->s[i].qty);
      VRF_MONEY(PSUPP, part->s[i].scost);
      VRF_STR(PSUPP, part->s[i].comment);
      VRF_END(PSUPP);
      }

   return(0);
}

/*
 * print the given part *and* its suppliers
 */
int
vrf_part_psupp(part_t *part, int mode)
{
    vrf_part(part, mode);
    vrf_psupp(part, mode);

    return(0);
}

int
vrf_supp(supplier_t *supp, int mode)
{
   VRF_STRT(SUPP);
   VRF_INT(SUPP, supp->suppkey);
   VRF_STR(SUPP, supp->name);
   VRF_STR(SUPP, supp->address);
   VRF_INT(SUPP, supp->nation_code);
   VRF_STR(SUPP, supp->phone);
   VRF_MONEY(SUPP, supp->acctbal);
   VRF_STR(SUPP, supp->comment);
   VRF_END(SUPP);

   return(0);
}

int
vrf_nation(code_t *c, int mode)
{
   VRF_STRT(NATION);
   VRF_INT(NATION, c->code);
   VRF_STR(NATION, c->text);
   VRF_INT(NATION, c->join);
   VRF_STR(NATION, c->comment);
   VRF_END(NATION);

   return(0);
}

int
vrf_region(code_t *c, int mode)
{
   VRF_STRT(REGION);
   VRF_INT(REGION, c->code);
   VRF_STR(REGION, c->text);
   VRF_STR(REGION, c->comment);
   VRF_END(fp);

   return(0);
}
```

==========

load_stub.c

```
==================================================
/*************************************************
************
 *  Title:      load_stub.c
 *  Sccsid:     @(#)load_stub.c      2.1.8.1
 *  Description:
 *              stub routines for:
 *          inline load of dss benchmark
 *          header creation for dss benchmark
 *
 *************************************************
************
 */

#include <stdio.h>
#include "config.h"
#include "dss.h"
#include "dsstypes.h"

int
close_direct(void)
{
    /* any post load cleanup goes here */
    return(0);
}

int
prep_direct(void)
{
    /* any preload prep goes here */
    return(0);
}

int
hd_cust (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
customer table\n");

    return(0);
}

int
ld_cust (customer_t *cp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the customer table");

    return(0);
}

int
hd_part (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
part table\n");

    return(0);
}

int
ld_part (part_t *pp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("No load routine has been defined for
the part table\n");
```

```
    return(0);
}

int
ld_psupp (part_t *pp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined for
the",
            "psupp table\n");

    return(0);

}


int
hd_supp (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
supplier table\n");

    return(0);
}

int
ld_supp (supplier_t *sp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the supplier table\n");

    return(0);
}


int
hd_order (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
order table\n");

    return(0);
}

int
ld_order (order_t *p, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the order table");

    return(0);
}

ld_line (order_t *p, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the line table");

    return(0);
}



int
hd_psupp (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No header has been defined for the",
            "part supplier table");

    return(0);
}


int
hd_line (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
lineitem table\n");

    return(0);
}

int
hd_nation (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
nation table\n");

    return(0);
}

int
ld_nation (code_t *cp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the nation table");

    return(0);
}

int
hd_region (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for the
region table\n");

    return(0);
}

int
ld_region (code_t *cp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("%s %s\n",
            "No load routine has been defined",
            "for the region table");

    return(0);
}

int
ld_order_line (order_t *p, int mode)
{
    ld_order(p, mode);
    ld_line (p, mode);

    return(0);
}

int
hd_order_line (FILE *f)
{
    hd_order(f);
    hd_line (f);

    return(0);
}

int
ld_part_psupp (part_t *p, int mode)
```

```
{
    ld_part(p, mode);
    ld_psupp (p, mode);

    return(0);
}

int
hd_part_psupp (FILE *f)
{
    hd_part(f);
    hd_psupp(f);

    return(0);
}
```

==========

build.c

```
==========
/* @(#)build.c 2.1.8.1 */
/* Sccsid:     @(#)build.c    9.1.1.17     11/15/95
12:52:28 */
/* stuff related to the customer table */
#include <stdio.h>
#include <string.h>
#ifndef VMS
#include <sys/types.h>
#endif
#if defined(SUN)
#include <unistd.h>
#endif
#include <math.h>

#include "dss.h"
#include "dsstypes.h"
#include "bcd2.h"
#ifdef ADHOC
#include "adhoc.h"
extern adhoc_t adhocs[];
#endif /* ADHOC */

#define LEAP_ADJ(yr, mnth)       \
((LEAP(yr) && (mnth) >= 2) ? 1 : 0)
#define JDAY_BASE       8035    /* start from 1/1/70 a
la unix */
#define JMNTH_BASE      (-70 * 12) /* start from
1/1/70 a la unix */
#define JDAY(date) ((date) - STARTDATE + JDAY_BASE +
1)
#define PART_SUPP_BRIDGE(tgt, p, s) \
    { \
    long tot_scnt = tdefs[SUPP].base * scale; \
    tgt = (p + s *  (tot_scnt / SUPP_PER_PART +  \
        (long) ((p - 1) / tot_scnt))) % tot_scnt + 1;
\
    }
#define RPRICE_BRIDGE(tgt, p) tgt = rpb_routine(p)
#define V_STR(avg, sd, tgt)  a_rnd((int)(avg *
V_STR_LOW), \
(int)(avg * V_STR_HGH), sd, tgt)
#define TEXT(avg, sd, tgt)  \
dbg_text(tgt, (int)(avg * V_STR_LOW),(int)(avg *
V_STR_HGH), sd)
static void gen_phone PROTO((long ind, char *target,
long seed));

long
rpb_routine(long p)
        {
        long price;

        price = 90000;
        price += (p/10) % 20001;        /* limit
contribution to $200 */
        price += (p % 1000) * 100;

        return(price);
        }

static void
gen_phone(long ind, char *target, long seed)
        {
    long       acode,
               exchg,
               number;

    RANDOM(acode, 100, 999, seed);
    RANDOM(exchg, 100, 999, seed);
        RANDOM(number, 1000, 9999, seed);
        sprintf(target, "%02d", 10 + (ind %
NATIONS_MAX));
    sprintf(target + 3, "%03d", acode);
    sprintf(target + 7, "%03d", exchg);
    sprintf(target + 11, "%04d", number);
    target[2] = target[6] = target[10] = '-';

    return;
        }



long
mk_cust(long n_cust, customer_t *c)
        {
        long i;

        c->custkey = n_cust;
        sprintf(c->name, C_NAME_FMT, C_NAME_TAG,
n_cust);
        c->alen = V_STR(C_ADDR_LEN, C_ADDR_SD,
c->address);
        RANDOM(i, 0, (nations.count - 1), C_NTRG_SD);
        c->nation_code = i;
        gen_phone(i, c->phone, (long)C_PHNE_SD);
        RANDOM(c->acctbal, C_ABAL_MIN, C_ABAL_MAX,
C_ABAL_SD);
        pick_str(&c_mseg_set, C_MSEG_SD,
c->mktsegment);
        c->clen = TEXT(C_CMNT_LEN, C_CMNT_SD,
c->comment);

        return (0);
        }


        /*
        * generate the numbered order and its
associated lineitems
*/
void
mk_sparse (DSS_HUGE i, DSS_HUGE *ok, long seq)
        {
#ifndef SUPPORT_64BITS
        if (scale < MAX_32B_SCALE)
#endif
                ez_sparse(i, ok, seq);
#ifndef SUPPORT_64BITS
        else
                hd_sparse(i, ok, seq);
#endif
        return;
        }

        /*
        * the "simple" version of mk_sparse, used on
systems with 64b support
        * and on all systems at SF <= 300G where 32b
support is sufficient
*/
void
ez_sparse(DSS_HUGE i, DSS_HUGE *ok, long seq)
        {
        long low_bits;
        LONG2HUGE(i, ok);
        low_bits = (long)(i & ((1 << SPARSE_KEEP) -
1));
        *ok = *ok >> SPARSE_KEEP;
        *ok = *ok << SPARSE_BITS;
        *ok += seq;
        *ok = *ok << SPARSE_KEEP;
        *ok += low_bits;


        return;
        }

#ifndef SUPPORT_64BITS
void
hd_sparse(long i, DSS_HUGE *ok, long seq)
        {
        long low_mask, seq_mask;
        static int init = 0;
        static DSS_HUGE *base, *res;

        if (init == 0)
                {
                INIT_HUGE(base);
                INIT_HUGE(res);
```

```
                init = 1;
                }

        low_mask = (1 << SPARSE_KEEP) - 1;
        seq_mask = (1 << SPARSE_BITS) - 1;
        bin_bcd2(i, base, base + 1);
        HUGE_SET (base, res);
        HUGE_DIV (res, 1 << SPARSE_KEEP);
        HUGE_MUL (res, 1 << SPARSE_BITS);
        HUGE_ADD (res, seq, res);
        HUGE_MUL (res, 1 << SPARSE_KEEP);
        HUGE_ADD (res, *base & low_mask, res);
        bcd2_bin (&low_mask, *res);
        bcd2_bin (&seq_mask, *(res + 1));
        *ok = low_mask;
        *(ok + 1) = seq_mask;
        return;
        }
#endif

long
mk_order(DSS_HUGE index, order_t *o, long upd_num)
        {
        long      lcnt;
        long      rprice;
        long      ocnt;
        long      tmp_date;
        long      s_date;
        long      r_date;
        long      c_date;
        long      clk_num;
        long      supp_num;
        static char **asc_date = NULL;
        char tmp_str[2];
        char **mk_ascdate PROTO((void));
        int delta = 1;

        if (asc_date == NULL)
         asc_date = mk_ascdate();
        mk_sparse (index, o->okey,
                (upd_num == 0) ? 0 : 1 + upd_num /
(10000 / refresh));
        RANDOM(o->custkey, O_CKEY_MIN, O_CKEY_MAX,
O_CKEY_SD);
    while (o->custkey % CUST_MORTALITY == 0)
                {
                o->custkey += delta;
                o->custkey = MIN(o->custkey,
O_CKEY_MAX);
                delta *= -1;
                }


    RANDOM(tmp_date, O_ODATE_MIN, O_ODATE_MAX,
O_ODATE_SD);
    strcpy(o->odate, asc_date[tmp_date - STARTDATE]);

    pick_str(&o_priority_set, O_PRIO_SD,
o->opriority);
        RANDOM(clk_num, 1, MAX((scale * O_CLRK_SCL),
O_CLRK_SCL), O_CLRK_SD);
    sprintf(o->clerk, O_CLRK_FMT,
         O_CLRK_TAG,
         clk_num);
    o->clen = TEXT(O_CMNT_LEN, O_CMNT_SD, o->comment);
#ifdef DEBUG
        if (o->clen > O_CMNT_MAX) fprintf(stderr,
"comment error: O%d\n", index);
#endif /* DEBUG */
    o->spriority = 0;

    o->totalprice = 0;
    o->orderstatus = 'O';
    ocnt = 0;

        RANDOM(o->lines, O_LCNT_MIN, O_LCNT_MAX,
O_LCNT_SD);
    for (lcnt = 0; lcnt < o->lines; lcnt++)
                {
         HUGE_SET(o->okey, o->l[lcnt].okey);
        o->l[lcnt].lcnt = lcnt + 1;
                RANDOM(o->l[lcnt].quantity, L_QTY_MIN,
L_QTY_MAX, L_QTY_SD);
                RANDOM(o->l[lcnt].discount, L_DCNT_MIN,
L_DCNT_MAX, L_DCNT_SD);
         RANDOM(o->l[lcnt].tax, L_TAX_MIN, L_TAX_MAX,
L_TAX_SD);
         pick_str(&l_instruct_set, L_SHIP_SD,
o->l[lcnt].shipinstruct);
         pick_str(&l_smode_set, L_SMODE_SD,
o->l[lcnt].shipmode);
         o->l[lcnt].clen = TEXT(L_CMNT_LEN, L_CMNT_SD,
o->l[lcnt].comment);
         RANDOM(o->l[lcnt].partkey, L_PKEY_MIN,
L_PKEY_MAX, L_PKEY_SD);
         RPRICE_BRIDGE( rprice, o->l[lcnt].partkey);
         RANDOM(supp_num, 0, 3, L_SKEY_SD);
         PART_SUPP_BRIDGE( o->l[lcnt].suppkey,
o->l[lcnt].partkey, supp_num);
         o->l[lcnt].eprice = rprice *
o->l[lcnt].quantity;

         o->totalprice +=
             ((o->l[lcnt].eprice *
             ((long)100 - o->l[lcnt].discount)) /
(long)PENNIES ) *
             ((long)100 + o->l[lcnt].tax)
             / (long)PENNIES;

                 RANDOM(s_date, L_SDTE_MIN, L_SDTE_MAX,
L_SDTE_SD);
         s_date += tmp_date;
                 RANDOM(c_date, L_CDTE_MIN, L_CDTE_MAX,
L_CDTE_SD);
         c_date += tmp_date;
                 RANDOM(r_date, L_RDTE_MIN, L_RDTE_MAX,
L_RDTE_SD);
         r_date += s_date;


         strcpy(o->l[lcnt].sdate, asc_date[s_date -
STARTDATE]);
         strcpy(o->l[lcnt].cdate, asc_date[c_date -
STARTDATE]);
         strcpy(o->l[lcnt].rdate, asc_date[r_date -
STARTDATE]);


         if (julian(r_date) <= CURRENTDATE)
                         {
             pick_str(&l_rflag_set, L_RFLG_SD,
tmp_str);
             o->l[lcnt].rflag[0] = *tmp_str;
                         }
         else
             o->l[lcnt].rflag[0] = 'N';

         if (julian(s_date) <= CURRENTDATE)
                         {
             ocnt++;
             o->l[lcnt].lstatus[0] = 'F';
                         }
         else
             o->l[lcnt].lstatus[0] = 'O';
                }

    if (ocnt > 0)
        o->orderstatus = 'P';
    if (ocnt == o->lines)
        o->orderstatus = 'F';

        return (0);
}

long
mk_part(long index, part_t *p)
        {
        long      temp;
        long      snum;
        long      brnd;

        p->partkey = index;
        agg_str(&colors, (long)P_NAME_SCL,
(long)P_NAME_SD, p->name);
        RANDOM(temp, P_MFG_MIN, P_MFG_MAX, P_MFG_SD);
        sprintf(p->mfgr, P_MFG_FMT, P_MFG_TAG, temp);
        RANDOM(brnd, P_BRND_MIN, P_BRND_MAX,
P_BRND_SD);
        sprintf(p->brand, P_BRND_FMT,
                P_BRND_TAG,
                (temp * 10 + brnd));
        p->tlen = pick_str(&p_types_set, P_TYPE_SD,
p->type);
        p->tlen =
strlen(p_types_set.list[p->tlen].text);
        RANDOM(p->size, P_SIZE_MIN, P_SIZE_MAX,
P_SIZE_SD);
        pick_str(&p_cntr_set, P_CNTR_SD,
p->container);
        RPRICE_BRIDGE( p->retailprice, index);
```

```
        p->clen = TEXT(P_CMNT_LEN, P_CMNT_SD,
p->comment);

        for (snum = 0; snum < SUPP_PER_PART; snum++)
                {
                p->s[snum].partkey = p->partkey;
                PART_SUPP_BRIDGE( p->s[snum].suppkey,
index, snum);
                RANDOM(p->s[snum].qty, PS_QTY_MIN,
PS_QTY_MAX, PS_QTY_SD);
                RANDOM(p->s[snum].scost, PS_SCST_MIN,
PS_SCST_MAX, PS_SCST_SD);
                p->s[snum].clen = TEXT(PS_CMNT_LEN,
PS_CMNT_SD, p->s[snum].comment);
                }
        return (0);
        }

long
mk_supp(long index, supplier_t *s)
        {
        long     i,
                bad_press,
                noise,
                offset,
                type;

        s->suppkey = index;
        sprintf(s->name, S_NAME_FMT, S_NAME_TAG,
index);
        s->alen = V_STR(S_ADDR_LEN, S_ADDR_SD,
s->address);
        RANDOM(i, 0, nations.count - 1, S_NTRG_SD);
        s->nation_code= i;
        gen_phone(i, s->phone, S_PHNE_SD);
        RANDOM(s->acctbal, S_ABAL_MIN, S_ABAL_MAX,
S_ABAL_SD);

        s->clen = TEXT(S_CMNT_LEN, S_CMNT_SD,
s->comment);
        /* these calls should really move inside the
if stmt below,
        * but this will simplify seedless parallel
load
        */
        RANDOM(bad_press, 1, 10000, BBB_CMNT_SD);
        RANDOM(type, 0, 100, BBB_TYPE_SD);
        RANDOM(noise, 0, (s->clen - BBB_CMNT_LEN),
BBB_JNK_SD);
        RANDOM(offset, 0, (s->clen - (BBB_CMNT_LEN +
noise)),
                BBB_OFFSET_SD);
        if (bad_press <= S_CMNT_BBB)
                {
                type = (type < BBB_DEADBEATS) ?0:1;
         memcpy(s->comment + offset, BBB_BASE,
BBB_BASE_LEN);
         if (type == 0)
                        memcpy(s->comment +
BBB_BASE_LEN + offset + noise,
                        BBB_COMPLAIN, BBB_TYPE_LEN);
         else
                        memcpy(s->comment +
BBB_BASE_LEN + offset + noise,
                        BBB_COMMEND, BBB_TYPE_LEN);
                }

        return (0);
        }

struct
        {
        char     *mdes;
        long      days;
        long      dcnt;
        }         months[] =

        {
                {NULL, 0, 0},
                {"JAN", 31, 31},
                {"FEB", 28, 59},
                {"MAR", 31, 90},
                {"APR", 30, 120},
                {"MAY", 31, 151},
                {"JUN", 30, 181},
                {"JUL", 31, 212},
                {"AUG", 31, 243},
                {"SEP", 30, 273},
                {"OCT", 31, 304},
                {"NOV", 30, 334},
                {"DEC", 31, 365}
                };

        long
                mk_time(long index, dss_time_t *t)
                {
                long      m = 0;
                long      y;
                long      d;

                t->timekey = index + JDAY_BASE;
                y = julian(index + STARTDATE - 1) /
1000;
                d = julian(index + STARTDATE - 1) %
1000;
                while (d > months[m].dcnt + LEAP_ADJ(y,
m))
                        m++;
                PR_DATE(t->alpha, y, m,
                        d - months[m - 1].dcnt -
((LEAP(y) && m > 2) ? 1 : 0));
                t->year = 1900 + y;
                t->month = m + 12 * y + JMNTH_BASE;
                t->week = (d + T_START_DAY - 1) / 7 +
1;
                t->day = d - months[m - 1].dcnt -
LEAP_ADJ(y, m-1);

                return (0);
                }

        int
                mk_nation(long index, code_t *c)
                {
                c->code = index - 1;
                c->text = nations.list[index - 1].text;
                c->join = nations.list[index -
1].weight;
                c->clen = TEXT(N_CMNT_LEN, N_CMNT_SD,
c->comment);
                return(0);
                }

        int
                mk_region(long index, code_t *c)
                {

                c->code = index - 1;
                c->text = regions.list[index - 1].text;
                c->join = 0;        /* for completeness
*/
                c->clen = TEXT(R_CMNT_LEN, R_CMNT_SD,
c->comment);
                return(0);
                }
```

dss.h

```
*/
/*
 * Sccsid:     @(#)dss.h      2.1.8.5
 *
 * general definitions and control information for the
DSS code
 * generator; if it controls the data set, it's here
 */
#ifndef DSS_H
#define  DSS_H
#ifdef TPCH
#define NAME                    "TPC-H"
#define VERSION           1
#define RELEASE           2
#define MODIFICATION      0
#define PATCH             ""
#endif
#ifdef TPCR
#define NAME                    "TPC-R"
#define VERSION           1
#define RELEASE           1
#define MODIFICATION      0
#define PATCH             ""
#endif
#ifndef NAME
#error Benchmark version must be defined in config.h
#endif
#define TPC             "Transaction Processing
Performance Council"
#define C_DATES         "1994 - 1999"
```

```
#include "config.h"
#include "shared.h"

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

#define  NONE          -1
#define  PART          0
#define  PSUPP         1
#define  SUPP          2
#define  CUST          3
#define  ORDER         4
#define  LINE          5
#define  ORDER_LINE 6
#define  PART_PSUPP 7
#define  NATION               8
#define  REGION               9
#define  UPDATE               10
#define  MAX_TABLE     11
#define  ONE_STREAM    1
#define  ADD_AT_END    2

#ifdef MAX
#undef MAX
#endif
#ifdef MIN
#undef MIN
#endif
#define MAX(a,b) ((a > b )?a:b)
#define MIN(A,B)  ( (A) < (B) ? (A) : (B))

#define INTERNAL_ERROR(p)  {fprintf(stderr,"%s",
p);abort();}
#define LN_CNT  4
static char lnoise[4] = {'|', '/', '-', '\\' };
#define LIFENOISE(n, var)       \
       if (verbose > 0) fprintf(stderr, "%c\b",
lnoise[(var%LN_CNT)])

#define MALLOC_CHECK(var) \
    if ((var) == NULL) \
        { \
        fprintf(stderr, "Malloc failed at %s:%d\n",  \
            __FILE__, __LINE__); \
        exit(1);\
        }
#define OPEN_CHECK(var, path) \
    if ((var) == NULL) \
        { \
        fprintf(stderr, "Open failed for %s at
%s:%d\n",  \
            path, __FILE__, __LINE__); \
        exit(1);\
        }
#ifndef MAX_CHILDREN
#define MAX_CHILDREN    1000
#endif

/*
 * macros that control sparse keys
 *
 * refer to Porting.Notes for a complete explanation
 */
#ifndef BITS_PER_LONG
#define BITS_PER_LONG   32
#define MAX_LONG        0x7FFFFFFF
#endif /* BITS_PER_LONG */
#define SPARSE_BITS      2
#define SPARSE_KEEP      3
#define  MK_SPARSE(key, seq) \
         (((((key>>3)<<2)|(seq & 0x0003))<<3)|(key &
0x0007))

#define RANDOM(tgt, lower, upper, stream)
       dss_random(&tgt, lower, upper, stream)



typedef struct
{
   long      weight;
   char     *text;
}         set_member;

typedef struct
{
   int      count;
   int      max;
   set_member *list;
   long *permute;
}         distribution;
/*
 * some handy access functions
 */
#define DIST_SIZE(d)          d->count
#define DIST_MEMBER(d, i)     ((set_member
*)((d)->list + i))->text

typedef struct
{
   char     *name;
   char     *comment;
   long      base;
   int       (*header) ();
   int       (*loader[2]) ();
   long      (*gen_seed)();
   int       (*verify) ();
   int       child;
   unsigned long vtotal;
}         tdef;

typedef struct SEED_T {
        long table;
        long value;
        long usage;
        long boundary;
        } seed_t;


#if defined(__STDC__)
#define PROTO(s) s
#else
#define PROTO(s) ()
#endif

/* bm_utils.c */
char    *env_config PROTO((char *var, char *dflt));
long    yes_no PROTO((char *prompt));
int     a_rnd PROTO((int min, int max, int column,
char *dest));
int     tx_rnd PROTO((long min, long max, long column,
char *tgt));
long    julian PROTO((long date));
long    unjulian PROTO((long date));
FILE    *tbl_open PROTO((int tbl, char *mode));
long    dssncasecmp PROTO((char *s1, char *s2, int
n));
long    dsscasecmp PROTO((char *s1, char *s2));
int             pick_str PROTO((distribution * s, int
c, char *target));
void    agg_str PROTO((distribution *set, long count,
long col, char *dest));
void    read_dist PROTO((char *path, char *name,
distribution * target));
void    embed_str PROTO((distribution *d, int min, int
max, int stream, char *dest));
#ifndef STDLIB_HAS_GETOPT
int             getopt PROTO((int arg_cnt, char
**arg_vect, char *oprions));
#endif /* STDLIB_HAS_GETOPT */
long    set_state PROTO((int t, long scale, long
procs, long step, long *e));

/* rnd.c */
long    NextRand PROTO((long nSeed));
long    UnifInt PROTO((long nLow, long nHigh, long
nStream));
double  UnifReal PROTO((double dLow, double dHigh,
long nStream));
double  Exponential PROTO((double dMean, long
nStream));
void    dss_random(long *tgt, long min, long max, long
seed);
void    row_start(int t);
void    row_stop(int t);
void    dump_seeds(int t);

/* text.c */
#define MAX_GRAMMAR_LEN       12     /* max length of
grammar component */
#define MAX_SENT_LEN   256 /* max length of populated
sentence */
#define RNG_PER_SENT   27     /* max number of RNG
calls per sentence */

int             dbg_text PROTO((char * t, int min, int
max, int s));

#ifdef DECLARER
```

```
#define EXTERN
#else
#define EXTERN extern
#endif            /* DECLARER */


EXTERN distribution nations;
EXTERN distribution nations2;
EXTERN distribution regions;
EXTERN distribution o_priority_set;
EXTERN distribution l_instruct_set;
EXTERN distribution l_smode_set;
EXTERN distribution l_category_set;
EXTERN distribution l_rflag_set;
EXTERN distribution c_mseg_set;
EXTERN distribution colors;
EXTERN distribution p_types_set;
EXTERN distribution p_cntr_set;

/* distributions that control text generation */
EXTERN distribution articles;
EXTERN distribution nouns;
EXTERN distribution adjectives;
EXTERN distribution adverbs;
EXTERN distribution prepositions;
EXTERN distribution verbs;
EXTERN distribution terminators;
EXTERN distribution auxillaries;
EXTERN distribution np;
EXTERN distribution vp;
EXTERN distribution grammar;


EXTERN long scale;
EXTERN int refresh;
EXTERN int resume;
EXTERN long verbose;
EXTERN long force;
EXTERN long header;
EXTERN long columnar;
EXTERN long direct;
EXTERN long updates;
EXTERN long table;
EXTERN long children;
EXTERN long fnames;
EXTERN int  gen_sql;
EXTERN int  gen_rng;
EXTERN char *db_name;
EXTERN int  step;
EXTERN int     set_seeds;
EXTERN int  validate;
EXTERN char *d_path;

/* added for segmented updates */
EXTERN int insert_segments;
EXTERN int delete_segments;
EXTERN int insert_orders_segment;
EXTERN int insert_lineitem_segment;
EXTERN int delete_segment;


#ifndef DECLARER
extern tdef tdefs[];

#endif            /* DECLARER */


/*************************************************
***********
 ** table level defines use the following naming
convention: t_ccc_xxx
 ** with: t, a table identifier
 **       ccc, a column identifier
 **       xxx, a limit type
 *************************************************
**********
 */

/*
 * defines which control the parts table
 */
#define  P_SIZE       126
#define  P_NAME_SCL   5
#define  P_MFG_TAG    "Manufacturer#"
#define  P_MFG_FMT     "%s%01d"
#define  P_MFG_MIN     1
#define  P_MFG_MAX     5
#define  P_BRND_TAG   "Brand#"
#define  P_BRND_FMT   "%s%02d"
#define  P_BRND_MIN     1
#define  P_BRND_MAX     5
#define  P_SIZE_MIN    1
#define  P_SIZE_MAX    50
#define  P_MCST_MIN    100
#define  P_MCST_MAX    99900
#define  P_MCST_SCL    100.0
#define  P_RCST_MIN    90000
#define  P_RCST_MAX    200000
#define  P_RCST_SCL    100.0
/*
 * defines which control the suppliers table
 */
#define  S_SIZE     145
#define  S_NAME_TAG "Supplier#"
#define  S_NAME_FMT "%s%09ld"
#define  S_ABAL_MIN   -99999
#define  S_ABAL_MAX    999999
#define  S_CMNT_MAX    101
#define  S_CMNT_BBB    10       /* number of BBB
comments/SF */
#define  BBB_DEADBEATS 50       /* % that are
complaints */
#define  BBB_BASE  "Customer "
#define  BBB_COMPLAIN  "Complaints"
#define  BBB_COMMEND   "Recommends"
#define  BBB_CMNT_LEN  19
#define  BBB_BASE_LEN  9
#define  BBB_TYPE_LEN  10

/*
 * defines which control the partsupp table
 */
#define  PS_SIZE      145
#define  PS_SKEY_MIN  0
#define  PS_SKEY_MAX  ((tdefs[SUPP].base - 1) * scale)
#define  PS_SCST_MIN  100
#define  PS_SCST_MAX  100000
#define  PS_QTY_MIN   1
#define  PS_QTY_MAX   9999
/*
 * defines which control the customers table
 */
#define  C_SIZE       165
#define  C_NAME_TAG   "Customer#"
#define  C_NAME_FMT   "%s%09d"
#define  C_MSEG_MAX    5
#define  C_ABAL_MIN   -99999
#define  C_ABAL_MAX    999999
/*
 * defines which control the order table
 */
#define  O_SIZE          109
#define  O_CKEY_MIN      1
#define  O_CKEY_MAX      (long)(tdefs[CUST].base *
scale)
#define  O_ODATE_MIN     STARTDATE
#define  O_ODATE_MAX     (STARTDATE + TOTDATE - \
                         (L_SDTE_MAX + L_RDTE_MAX) -
1)
#define  O_CLRK_TAG      "Clerk#"
#define  O_CLRK_FMT      "%s%09d"
#define  O_CLRK_SCL      1000
#define  O_LCNT_MIN      1
#define  O_LCNT_MAX      7

/*
 * defines which control the lineitem table
 */
#define  L_SIZE       144L
#define  L_QTY_MIN    1
#define  L_QTY_MAX    50
#define  L_TAX_MIN    0
#define  L_TAX_MAX    8
#define  L_DCNT_MIN   0
#define  L_DCNT_MAX   10
#define  L_PKEY_MIN   1
#define  L_PKEY_MAX   (tdefs[PART].base * scale)
#define  L_SDTE_MIN   1
#define  L_SDTE_MAX   121
#define  L_CDTE_MIN   30
#define  L_CDTE_MAX   90
#define  L_RDTE_MIN   1
#define  L_RDTE_MAX   30
/*
 * defines which control the time table
 */
#define  T_SIZE       30
#define  T_START_DAY  3     /* wednesday ? */
#define  LEAP(y)  ((!(y % 4) && (y % 100))?1:0)
```

```
/*************************************************
*************
 *************************************************
*************
 ***
 *** general or inter table defines
 ***
 *************************************************
*************
 *************************************************
*************/
#define  SUPP_PER_PART 4
#define  ORDERS_PER_CUST 10 /* sync this with
CUST_MORTALITY */
#define  CUST_MORTALITY 3  /* portion with have no
orders */
#define  NATIONS_MAX  90 /* limited by country codes
in phone numbers */
#define  PHONE_FMT    "%02d-%03d-%03d-%04d"
#define  STARTDATE    92001
#define  CURRENTDATE  95168
#define  ENDDATE      98365
#define  TOTDATE      2557
#define  UPD_PCT      10
#define  MAX_STREAM   47
#define  V_STR_LOW    0.4
#define  PENNIES    100 /* for scaled int money
arithmetic */
#define  Q11_FRACTION (double)0.0001
/*
 * max and min SF in GB; Larger SF will require
changes to the build routines
 */
#define  MIN_SCALE      1.0
#define  MAX_SCALE   3000.0
/*
 * beyond this point we need to allow for BCD
calculations
 */
#define  MAX_32B_SCALE   3000.0
#define INIT_HUGE(v)   { \
                       v = (DSS_HUGE
*)malloc(sizeof(DSS_HUGE) * HUGE_COUNT); \
                       MALLOC_CHECK(v); \
                       }
#define FREE_HUGE(v)   free(v)
#ifdef SUPPORT_64BITS
#define LONG2HUGE(src, dst)          *dst =
(DSS_HUGE)src
#define HUGE2LONG(src, dst)          *dst = (long)src
#define HUGE_SET(src, dst)           *dst = *src
#define HUGE_MUL(op1, op2)           *op1 *= op2
#define HUGE_DIV(op1, op2)           *op1 /= op2
#define HUGE_ADD(op1, op2, dst)      *dst = *op1 +
op2
#define HUGE_SUB(op1, op2, dst)      *dst = *op1 -
op2
#define HUGE_MOD(op1, op2)           *op1 % op2
#define HUGE_CMP(op1, op2)           (*op1 ==
*op2)?0:(*op1 < *op2)-1:1
#else
#define LONG2HUGE(src, dst)          {*dst = src;
*(dst + 1) = 0;}
#define HUGE2LONG(src, dst)          { dst=0 ; \
                                     bcd2_bin(dst,
(src + 1)); \
                                     bcd2_bin(dst,
src); }
#define HUGE_SET(src, dst)           { *dst = *src ;
*(dst + 1) = *(src + 1); }
#define HUGE_MUL(op1,op2)            bcd2_mul(op1,
(op1 + 1), op2)
#define HUGE_DIV(op1,op2)            bcd2_div(op1,
(op1 + 1), op2)
#define HUGE_ADD(op1,op2,d)          { \
                                     HUGE_SET(op1,
d); \
                                     bcd2_add(d, (d +
1), op2); \
                                     }
#define HUGE_SUB(op1,op2,d)          { \
                                     HUGE_SET(op1,
d); \
                                     bcd2_sub(d, (d +
1), op2); \
                                     }
#define HUGE_MOD(op1, op2)           bcd2_mod(op1,
(op1 + 1), op2)
#define HUGE_CMP(op1, op2)           (bcd2_cmp(op1,
(op1 + 1), op2) == 0)?0:\
((bcd2_cmp(op1, (op1 + 1), op2) < 0)?-1:1)
#endif /* SUPPORT_64BITS */

/******** environmental variables and defaults
**************/
#define  DIST_TAG  "DSS_DIST"        /* environment
var to override ... */
#define  DIST_DFLT "dists.dss"               /*
default file to hold distributions */
#define  PATH_TAG  "DSS_PATH"        /* environment
var to override ... */
#define  PATH_DFLT "."
       /* default directory to hold tables */
#define  CONFIG_TAG  "DSS_CONFIG"    /* environment
var to override ... */
#define  CONFIG_DFLT "."                     /*
default directory to config files */
#define  ADHOC_TAG  "DSS_ADHOC"              /*
environment var to override ... */
#define  ADHOC_DFLT "adhoc.dss"              /*
default file name for adhoc vars */

/******* output macros ********/
#ifndef SEPARATOR
#define SEPARATOR '|' /* field spearator for generated
flat files */
#endif
/* Data type flags for a single print routine */
#define DT_STR        0
#ifndef MVS
#define DT_VSTR               DT_STR
#else
#define DT_VSTR               1
#endif /* MVS */
#define DT_INT        2
#define DT_HUGE               3
#define DT_KEY        4
#define DT_MONEY      5
#define DT_CHR        6

int dbg_print(int dt, FILE *tgt, void *data, int len,
int eol);
#define PR_STR(fp, str, len)
       dbg_print(DT_STR, fp, (void *)str, len, 1)
#define PR_VSTR(fp, str, len)
       dbg_print(DT_VSTR, fp, (void *)str, len, 1)
#define PR_VSTR_LAST(fp, str, len)
       dbg_print(DT_VSTR, fp, (void *)str, len, 0)
#define PR_INT(fp, str)
       dbg_print(DT_INT, fp, (void *)str, 0, 1)
#define PR_HUGE(fp, str)
       dbg_print(DT_HUGE, fp, (void *)str, 0, 1)
#define PR_KEY(fp, str)
       dbg_print(DT_KEY, fp, (void *)str, 0, -1)
#define PR_MONEY(fp, str)
       dbg_print(DT_MONEY, fp, (void *)str, 0, 1)
#ifdef LINUX
  #define PR_CHR(fp, str)
       dbg_print(DT_CHR, fp, str, 0, 1)
#else
  #define PR_CHR(fp, str)
       dbg_print(DT_CHR, fp, (void *)str, 0, 1)
#endif
#define  PR_STRT(fp)   /* any line prep for a record
goes here */
#define  PR_END(fp)    fprintf(fp, "\n")   /* finish
the record here */
#ifdef MDY_DATE
#define  PR_DATE(tgt, yr, mn, dy)     \
   sprintf(tgt, "%02d-%02d-19%02d", mn, dy, yr)
#else
#define  PR_DATE(tgt, yr, mn, dy)     \
sprintf(tgt, "19%02d-%02d-%02d", yr, mn, dy)
#endif /* DATE_FORMAT */

/*
 * verification macros
 */
#define  VRF_STR(t, d) {char *xx = d; while (*xx)
tdefs[t].vtotal += *xx++;}
#define  VRF_INT(t,d)  tdefs[t].vtotal += d
#ifdef SUPPORT_64BITS
#define  VRF_HUGE(t,d)        tdefs[t].vtotal =
*((long *)&d) + *((long *)(&d + 1))
#else
#define VRF_HUGE(t,d)  tdefs[t].vtotal += d[0] + d[1]
#endif /* SUPPORT_64BITS */
/* assume float is a 64 bit quantity */
#define  VRF_MONEY(t,d)       tdefs[t].vtotal =
```

```
*((long *)&d) + *((long *)(&d + 1))
#define  VRF_CHR(t,d)  tdefs[t].vtotal += d
#define  VRF_STRT(t)
#define  VRF_END(t)

/*********** distribuitons currently defined
*************/
#define  UNIFORM   0

/*
 * seed indexes; used to separate the generation of
individual columns
 */
#define  P_MFG_SD  0
#define  P_BRND_SD 1
#define  P_TYPE_SD 2
#define  P_SIZE_SD 3
#define  P_CNTR_SD 4
#define  P_RCST_SD 5
#define  PS_QTY_SD 7
#define  PS_SCST_SD   8
#define  O_SUPP_SD 10
#define  O_CLRK_SD 11
#define  O_ODATE_SD   13
#define  L_QTY_SD  14
#define  L_DCNT_SD 15
#define  L_TAX_SD  16
#define  L_SHIP_SD 17
#define  L_SMODE_SD   18
#define  L_PKEY_SD 19
#define  L_SKEY_SD 20
#define  L_SDTE_SD 21
#define  L_CDTE_SD 22
#define  L_RDTE_SD 23
#define  L_RFLG_SD 24
#define  C_NTRG_SD 27
#define  C_PHNE_SD 28
#define  C_ABAL_SD 29
#define  C_MSEG_SD 30
#define  S_NTRG_SD 33
#define  S_PHNE_SD 34
#define  S_ABAL_SD 35
#define  P_NAME_SD 37
#define  O_PRIO_SD 38
#define  HVAR_SD   39
#define  O_CKEY_SD 40
#define  N_CMNT_SD 41
#define  R_CMNT_SD 42
#define  O_LCNT_SD 43
#define  BBB_JNK_SD    44
#define  BBB_TYPE_SD   45
#define  BBB_CMNT_SD   46
#define  BBB_OFFSET_SD 47
#endif            /* DSS_H */
```

```
=====================================================
```

config.h

```
=====================================================
/*
 * Sccsid:     @(#)config.h   2.1.8.2
 *
 * this file allows the compilation of DBGEN to be
tailored to specific
 * architectures and operating systems. Some options
are grouped
 * together to allow easier compilation on a given
vendor's hardware.
 *
 * The following #defines will effect the code:
 *   TPCH              -- make will create TPCH (set
in makefile)
 *   TPCR              -- make will create TPCR (set
in makefile)
 *   KILL(pid)         -- how to terminate a process
in a parallel load
 *   SPAWN             -- name of system call to clone
an existing process
 *   SET_HANDLER(proc) -- name of routine to handle
signals in parallel load
 *   WAIT(res, pid)    -- how to await the termination
of a child
 *   SEPARATOR         -- character used to separate
fields in flat files
 *   DBNAME            -- default name of database to
be loaded
 *   STDLIB_HAS_GETOPT -- to prevent confilcts with
gloabal getopt()
 *   MDY_DATE          -- generate dates as MM-DD-YY
 *   WIN32             -- support for WindowsNT
 *   SUPPORT_64BITS    -- compiler defines a 64 bit
datatype
 *   DSS_HUGE          -- 64 bit data type
 *   HUGE_FORMAT       -- printf string for 64 bit
data type
 *   HUGE_COUNT        -- number of objects in
DSS_HUGE
 *   EOL_HANDLING      -- flat files don't need final
column separator
 *
 *   OS defines
 *   ==========
 *   ATT        -- getopt() handling
 *   DOS        -- disable all multi-user
functionality/dependency
 *   HP         -- posix source inclusion differences
 *   IBM        -- posix source inclusion differences
 *   ICL        -- getopt() handling
 *   MVS        -- special handling of varchar format
 *   SGI        -- getopt() handling
 *   SUN        -- getopt() handling
 *   TANDEM     -- EOL handling
 *   U2200      -- death of parent kills children
automatically
 *   VMS        -- signal/fork handing differences
 *
 *   Database defines
 *   ================
 *   DB2        -- use DB2 dialect in QGEN
 *   INFORMIX   -- use Informix dialect in QGEN
 *   SQLSERVER  -- use SQLSERVER dialect in QGEN
 *   SYBASE     -- use Sybase dialect in QGEN
 *   TDAT       -- use Teradata dialect in QGEN
 */

#ifdef DOS
#define DSS_PROC        1
#define PATH_SEP       '\\'
#else


#ifdef ATT
#define STDLIB_HAS_GETOPT
#ifdef SQLSERVER
#define WIN32
#else
/* the 64 bit defines are for the Metaware compiler */
#define SUPPORT_64BITS
#define DSS_HUGE long long
#define HUGE_COUNT    1
#define HUGE_FORMAT "%LLd"
#endif /* SQLSERVER or MP/RAS */
#endif /* ATT */

#ifdef HP
#define _INCLUDE_POSIX_SOURCE
#define STDLIB_HAS_GETOPT
#endif /* HP */

#ifdef IBM
#define _POSIX_SOURCE
/*
 * if the C compiler is 3.1 or later, then uncomment
the
 * lines for 64 bit seed generation
 */
/* #define SUPPORT_64BITS */
/* #define DSS_HUGE long long */
/* #define HUGE_COUNT  1 */
#define STDLIB_HAS_GETOPT
#endif /* IBM */

#ifdef ICL
#define STDLIB_HAS_GETOPT
#endif /* ICL */
#define SUN
#ifdef SUN
#define STDLIB_HAS_GETOPT
#define SUPPORT_64BITS
#define DSS_HUGE long long
#define HUGE_FORMAT "%lld"
#define HUGE_COUNT    1
#endif /* SUN */

#ifdef SGI
#define STDLIB_HAS_GETOPT
#define SUPPORT_64BITS
#define DSS_HUGE __uint64_t
#define HUGE_COUNT    1
#endif /* SGI */
```

```
#ifdef TANDEM
#define EOL_HANDLING
#endif /* TANDEM */

#ifdef VMS
#define SPAWN   vfork
#define KILL(pid) kill(SIGQUIT, pid)
#define SET_HANDLER(proc) signal(SIGQUIT, proc)
#define WAIT(res, pid) wait(res)
#define SIGS_DEFINED
#endif /* VMS */

#if (defined(WIN32)&&!defined(_POSIX_))
#define pid_t int
#define SET_HANDLER(proc) signal(SIGINT, proc)
#define KILL(pid) \

TerminateProcess(OpenProcess(PROCESS_TERMINATE,FALSE,p
id),3)
#if (defined (__WATCOMC__))
#define SPAWN()   spawnv(P_NOWAIT, spawn_args[0],
spawn_args)
#define WAIT(res, pid) cwait(res, pid, WAIT_CHILD)
#else
#define SPAWN()   _spawnv(_P_NOWAIT, spawn_args[0],
spawn_args)
#define WAIT(res, pid) _cwait(res, pid, _WAIT_CHILD)
#define getpid          _getpid
#endif /* WATCOMC */
#define SIGS_DEFINED
#define PATH_SEP       '\\'
#ifndef TEST_32B
#define SUPPORT_64BITS
#define DSS_HUGE __int64
#define HUGE_COUNT    1
#define HUGE_FORMAT "%I64d"
#endif /* TEST_32B */
/* need to define process termination codes to match
UNIX */
/* these are copied from Linux/GNU and need to be
verified as part of a rework of */
/* process handling under NT (29 Apr 98) */
#define WIFEXITED(s)  ((s & 0xFF) == 0)
#define WIFSIGNALED(s)        (((unsigned
int)((status)-1) & 0xFFFF) < 0xFF)
#define WIFSTOPPED(s)  (((s) & 0xff) == 0x7f)
#define WTERMSIG(s)           ((s) & 0x7f)
#define WSTOPSIG(s)           (((s) & 0xff00) >> 8)
#endif /* WIN32 */

#ifndef SIGS_DEFINED
#define KILL(pid) kill(SIGUSR1, pid)
#define SET_HANDLER(proc) signal(SIGUSR1, proc)
#define SPAWN   fork
#define WAIT(res, pid) wait(res)
#endif /* DEFAULT */

#define DSS_PROC        getpid()
#endif /* DOS */

#ifndef DBNAME
#define DBNAME "dss"
#endif /* DBNAME */

#ifndef PATH_SEP
#define PATH_SEP '/'
#endif /* PATH_SEP */

#ifndef DSS_HUGE
#define DSS_HUGE      long
#define HUGE_COUNT    2
#endif
```

==========

dsstypes.h

```
==================================================
*/
 /*
 * Sccsid:     @(#)dsstypes.h 2.1.8.1
 *
 * general definitions and control information for the
DSS data types
 * and function prototypes
 */

/*
 * typedefs
 */

typedef struct
{
    long            custkey;
    char            name[C_NAME_LEN + 1];
    char            address[C_ADDR_MAX + 1];
    int             alen;
    long            nation_code;
    char            phone[PHONE_LEN + 1];
    long            acctbal;
    char            mktsegment[MAXAGG_LEN + 1];
    char            comment[C_CMNT_MAX + 1];
    int             clen;
}              customer_t;
/* customers.c */
long mk_cust   PROTO((long n_cust, customer_t * c));
int pr_cust    PROTO((customer_t * c, int mode));
int ld_cust    PROTO((customer_t * c, int mode));

typedef struct
{
    DSS_HUGE       *okey;
    long            partkey;
    long            suppkey;
    long            lcnt;
    long            quantity;
    long            eprice;
    long            discount;
    long            tax;
    char            rflag[1];
    char            lstatus[1];
    char            cdate[DATE_LEN];
    char            sdate[DATE_LEN];
    char            rdate[DATE_LEN];
    char           shipinstruct[MAXAGG_LEN + 1];
    char           shipmode[MAXAGG_LEN + 1];
    char           comment[L_CMNT_MAX + 1];
    int            clen;
}              line_t;

typedef struct
{
    DSS_HUGE       *okey;
    long            custkey;
    char            orderstatus;
    long            totalprice;
    char            odate[DATE_LEN];
    char            opriority[MAXAGG_LEN + 1];
    char            clerk[O_CLRK_LEN + 1];
    long            spriority;
    long            lines;
    char            comment[O_CMNT_MAX + 1];
    int            clen;
    line_t          l[O_LCNT_MAX];
}              order_t;

/* order.c */
long    mk_order       PROTO((DSS_HUGE index, order_t
* o, long upd_num));
int            pr_order       PROTO((order_t * o, int
mode));
int            ld_order       PROTO((order_t * o, int
mode));
void    ez_sparse      PROTO((DSS_HUGE index, DSS_HUGE
*ok, long seq));
#ifndef SUPPORT_64BITS
void    hd_sparse      PROTO((long index, DSS_HUGE
*ok, long seq));
#endif


typedef struct
{
    long            partkey;
    long            suppkey;
    long            qty;
    long            scost;
    char           comment[PS_CMNT_MAX + 1];
    int            clen;
}              partsupp_t;

typedef struct
{
    long           partkey;
    char           name[P_NAME_LEN + 1];
    int            nlen;
    char           mfgr[P_MFG_LEN + 1];
    char           brand[P_BRND_LEN + 1];
    char           type[P_TYPE_LEN + 1];
    int            tlen;
    long           size;
```

```
    char           container[P_CNTR_LEN + 1];
    long           retailprice;
    char           comment[P_CMNT_MAX + 1];
    int            clen;
    partsupp_t     s[SUPP_PER_PART];
}               part_t;

/* parts.c */
long mk_part   PROTO((long index, part_t * p));
int pr_part    PROTO((part_t * part, int mode));
int ld_part    PROTO((part_t * part, int mode));

typedef struct
{
    long            suppkey;
    char            name[S_NAME_LEN + 1];
    char            address[S_ADDR_MAX + 1];
    int             alen;
    long            nation_code;
    char            phone[PHONE_LEN + 1];
    long            acctbal;
    char            comment[S_CMNT_MAX + 1];
    int             clen;
}               supplier_t;
/* supplier.c */
long mk_supp   PROTO((long index, supplier_t * s));
int pr_supp    PROTO((supplier_t * supp, int mode));
int ld_supp    PROTO((supplier_t * supp, int mode));

typedef struct
{
    long            timekey;
    char            alpha[DATE_LEN];
    long            year;
    long            month;
    long            week;
    long            day;
} dss_time_t;

/* time.c */
long mk_time   PROTO((long index, dss_time_t * t));

/*
 * this assumes that N_CMNT_LEN >= R_CMNT_LEN
 */
typedef struct
{
    long            code;
    char            *text;
    long            join;
    char            comment[N_CMNT_MAX + 1];
    int             clen;
}               code_t;

/* code table */
int mk_nation   PROTO((long i, code_t * c));
int pr_nation    PROTO((code_t * c, int mode));
int ld_nation    PROTO((code_t * c, int mode));
int mk_region   PROTO((long i, code_t * c));
int pr_region    PROTO((code_t * c, int mode));
int ld_region    PROTO((code_t * c, int mode));
```

==========

ACID Test Source Code

==========

a_query.sh

==========

```
#!/bin/ksh
ade useview acid2 << END
sqlplus tpcd/tpcd
@/private/tpcd/acid/kit/acid/isolation/a_query $1
exit
END
```

==========

a_query.sql

==========

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

set serverout on;

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

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

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

exit;
```

==========

a_query2.sh

==========

```
#!/bin/ksh
ade useview acid2 << END
sqlplus tpcd/tpcd
@/private/tpcd/acid/kit/acid/isolation/a_query2 $1 $2
exit
END
```

==========

a_query2.sql

==========

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

```
ps_suppkey
rem       The range for ps_partkey is 1 to 20000
rem       The range for ps_suppkey is 1 to 1000
rem       A valid combination is 46 and 47

set serverout on;

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

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

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

exit;
```

atom.sh

```
#!/bin/ksh
#
# $Header: atom.sh 08-aug-99.13:48:02 mpoess Exp $
#
# atom.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights
Reserved.
#
#    NAME
#      atom.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Performs atomicity tests.
#      Usage: atom.sh [-n iter] [-p prog] [-u
usr/pswd] -h
#
#      Options: See usage below
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/08/99 - Creation
#    mpoess      08/08/99 - Creation
#

. $KIT_DIR/env

OH=$ORACLE_HOME
# ACID_DIR=$TPCD_KIT_DIR/audit set in env
OUT_DIR=$ACID_OUT
DURA_DIR=$ACID_DIR/dura
SF=1

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=atranspl
OUT=${OUT_DIR}/atom
USER=tpcd/tpcd

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 ""

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 ""

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

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

atranspl.sh

```
#!/bin/ksh
ade useview acid2 << END
echo I am in atranspl.sh
echo my arguments are $1 $2 $3 $4 $5 $6 $7
/private/tpcd/acid/kit/utils/atranspl $1 $2 $3 $4 $5
$6 $7
exit
END
```

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

svrmgrl << !

       connect internal;
       alter system switch logfile;
       alter system switch logfile;
       exit;
!
```

cnt_hist.sql

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

```
exit;

====================================================
```

consist.sh

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

. $KIT_DIR/env

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

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

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

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

STREAM=9
ITER=100
PROG=atranspl
USER="tpcd/tpcd"
CK=10

usage() {

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

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

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


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

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

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

i=0

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

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

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

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

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

i=0

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

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

(sleep $CK; $ACID_DIR/consistency/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 /*+ index(lineitem,i_l_orderkey) */
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
```

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

ITER=1000
STEM=9
PROG=${ACID_DIR}/atranspl.ott
IN=${ACID_DIR}/acid_in
DURA=${DURA_DIR}/dura
DURA=${DURA_DIR}/drate
OUT=${DURA_DIR}/drate
DSMPL=${DURA_DIR}/durasmpl
KEY=${DURA_DIR}/key${1}_
USER=tpcd/tpcd
TRIG=1
HCNT=duracnta

# get history count

sqlplus $USER @cnt_hist > $DURA_DIR/$HCNT

# perform the consistency

i=0
while [ $i -lt $STEM ]
do
   for j in ‘head -10 ${KEY}${i} | awk ’{printf "%d
",$1}’‘
   do
      sqlplus tpcd/tpcd @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
```

iso.sh

```
=====================================================
#!/bin/ksh
#
# $Header: iso.sh 17-aug-99.15:44:51 mpoess Exp $
#
# iso.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights
Reserved.
#
#    NAME
#      iso.sh
#
#    DESCRIPTION
#     This script triggers all 6 isolation tests. In
addition,
#     it creates more readable formats of the
isolation test output.
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/17/99 - Creation
#    mpoess      08/17/99 - Creation
#

for iso in iso1 iso2 iso3 iso4 iso5 iso6;do
       echo Running isolation test $iso
       ${iso}.sh
       echo Creating nicely formated output of ACID
test $iso
       xiso.pl -o ${ACID_OUT}/${iso}
done

=====================================================
```

iso1.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh
#ADE="ade useview acid2;"

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

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

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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


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

# generate key files

randkey 1 1 u"$USER" > $KEYFILE
echo -----
ls -l $KEYFILE
cat $KEYFILE
echo ++++++
OKEY=`cat $KEYFILE | awk '{print $1}'`
echo "o_key is "$OKEY

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

echo "Running ACID query BEFORE the start of Isolation
Test 1" >> $TXN2FILE
echo "`date`" >> $TXN2FILE
echo "" >> $TXN2FILE
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
echo "" >> $TXN2FILE
echo
"-----------------------------------------------" >>
$TXN2FILE
sleep 1
echo before PROG
# start ACID transaction, Sleep for 60 second before
COMMIT

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

echo PROG 1 1 1 0 i$KEYFILE u$USER s60

# let's sleep 10 seconds before starting ACID query

sleep 10
# start ACID query with the same OKEY

echo "Running ACID query 10 seconds AFTER the start of
ACID Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
#${RSH} -n ${HOST}  sqlplus $USER
@$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
${RSH} -n ${HOST} a_query.sh $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

echo
"-----------------------------------------------" >>
$TXN2FILE
wait
echo
```

```
"-----------------------------------------------" >>
$TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

=====================================================
```

iso2.sh

```
=====================================================
#!/bin/ksh
#
# $Header: iso2.sh 04-aug-99.09:19:54 mpoess Exp $
#
# iso2.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights
Reserved.
#
#    NAME
#      iso2.sh - <one-line expansion of the name>
#
#    DESCRIPTION
#      Usage: iso2.sh [-u user/password] [-n
remote_node] -h
#      Options: See usage below
#    NOTES
#       For a cross node isolation test, assume the
local node is
#       one of the participating nodes.  The other
node can be
#       specified by the -n option.
#       You need to set the environment variable
TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#
#
#
======================================================
============+
# May need to change the following:

. $KIT_DIR/env

RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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

# generate key files

randkey 1 1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 30 second before
ROLLBACK

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

# let's sleep 10 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY

echo "Running ACID query 10 seconds AFTER the start of
ACID transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
#${RSH} -n ${HOST}  sqlplus $USER
@$ACID_DIR/isolarion/a_query $OKEY >> $TXN2FILE
${RSH} -n ${HOST} a_query.sh $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

=====================================================
```

iso3.sh

```
=====================================================
#!/bin/ksh
#
# $Header: iso3.sh 04-aug-99.09:20:35 mpoess Exp $
#
# iso3.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights
Reserved.
#
#    NAME
#      iso3.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Usage: iso3.sh [-u user/password] [-n
remote_node] -h
#      Options: See usage below
#  NOTES
#       For a cross node isolation test, assume the
local node is
#       one of the participating nodes.  The other
node can be
#       specified by the -n option.
```

```
#       We need to make sure the remote node has
access to the
#       file system on the local node.  Otherwise, we
need to rcp
#       the keyfile to the remote system.
#       You need to set the environment variable
TPCD_KIT_DIR
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/04/99 - Creation
#    mpoess      08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER="tpcd/tpcd"
PROG=$KIT_DIR/utils/atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 1 u"$USER" > $KEYFILE
k1=`cat $KEYFILE | awk '{print $1}'`
k2=`cat $KEYFILE | awk '{print $2}'`
k3=`cat $KEYFILE | awk '{print $3}'`
sleep 1

# start ACID transaction, Sleep for 30 second before
COMMIT

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

# let's sleep 10 seconds before starting second ACID
transaction

sleep 10

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

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

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST} atranspl.sh 2 2 1 1 $k1 $k2 $k3
u$USER s1 >> $TXN2FILE &
#${RSH} -n ${HOST}
/net/b33a/private/tpcd/acid/kit/acid/isolation/atransp
l.sh 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

==========

iso4.sh

==========

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

TXN1FILE=$OUT_DIR/txn1$$.out
TXN2FILE=$OUT_DIR/txn2$$.out
KEYFILE=/tpch_ff/ff_ps8/key$$.out
ISOFILE=$OUT_DIR/iso4

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 1 u"$USER" > $KEYFILE

k1=`cat $KEYFILE | awk '{print $1}'`
k2=`cat $KEYFILE | awk '{print $2}'`
k3=`cat $KEYFILE | awk '{print $3}'`


sleep 1

# start ACID transaction, Sleep for 30 second before
ROLLBACK

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

# let's sleep 10 seconds before starting second ACID
transaction

sleep 10

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

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

if [ "$HOST" != "" ]
then
echo "Starting TXN2 on node $HOST" >> $TXN2FILE
#${RSH} -n ${HOST} $PROG 2 2 1 1 i$KEYFILE u$USER s1
>> $TXN2FILE &
${RSH} -n ${HOST} atranspl.sh 2 2 1 1 $k1 $k2 $k3
u$USER s1 >> $TXN2FILE &
else
$PROG 2 2 1 1 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

======================================================
```

iso5.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

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

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before
COMMIT

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

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY
```

```
PSKEY=‘randpsup 0.1‘

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

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

wait

echo
"−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−" >>
$TXN2FILE
echo
"−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−" >>
$TXN1FILE

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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


==================================================
```

iso6.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER="tpcd/tpcd"
PROG=atranspl

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

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

usage() {

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

set −− ‘getopt "u:n:h" "$@"‘ || usage

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


# generate key files

randkey 1 1 u"$USER" > $KEYFILE

k1=‘cat $KEYFILE | awk ’{print $1}’‘
k2=‘cat $KEYFILE | awk ’{print $2}’‘
k3=‘cat $KEYFILE | awk ’{print $3}’‘


OKEY=‘cat $KEYFILE | awk ’{print $1}’‘
echo "o_key is "$OKEY

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

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

sleep 1

# start Query 17, use 0 as the delta
echo output of Query 1 goes to $TXN1FILE
echo "Running Query 17b at ‘date‘" >> $TXN1FILE
sqlplus $USER @q1 >> $TXN1FILE &

# sleep 2 seconds before starting ACID transaction

sleep 10

# start ACID transaction, COMMIT after one second

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

if [ "$HOST" != "" ]
then
echo "Starting ACID transaction on node $HOST" >>
$TXN2FILE
${RSH} −n ${HOST} atranspl.sh 1 1 1 0 $k1 $k2 $k3
u$USER s1 >> ${TXN2FILE} &
#${RSH} −n ${HOST} atranspl.sh 1 1 1 0 i$KEYFILE
u$USER s1 >> ${TXN2FILE} &
else
$PROG 1 1 1 0 i$KEYFILE u$USER s1 >> $TXN2FILE &
fi

echo
"−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−" >>
```

```
$TXN1FILE
sleep 240
echo TXN1FILE $TXN1FILE
echo TXN2FILE $TXN2FILE
echo TXN3FILE $TXN3FILE
cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE
#/bin/rm -rf $TXN1FILE $TXN2FILE $TXN3FILE $KEYFILE

=====================================================
```

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 tpcd/tpcd"
echo "-t trigger  : trigger time between process
starts, default is 1 second"
   echo "-h          : print this usage summary"
   exit 1;
}

ITER=1000
STEM=9
SF=1
PROG=atranspl
IN=${ACID_DIR}/acid_in
DURA_DIR=$ACID_OUT/dura
OUT=${DURA_DIR}/drate
DURA=${DURA_DIR}/dura
KEY=${DURA_DIR}/key$$_
USER=tpcd/tpcd
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 durability run..."

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

# Get history count before the run

sqlplus -s $USER << ! > $DURA_DIR/$HCNT 2>&1
connect tpcd/tpcd
@cnt_hist
!

while [ $i -lt $STEM ]
do
   randkey 1000 ${SF} u${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 tpcd/tpcd @consist $j >>
$DURA_DIR/duraconsb
   done
   i=`expr $i + 1`
done

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

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 tpcd/tpcd @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;
```

==========

Disk Configuration Details

==========

Basic Assumptions:

1. Use T3 with write cache for redo logs.
2. Use A5200 striped wide for everything else.

Disk Config:

1 - T3 rack w/ 8 bricks of 9 x 18G drives.
10 - A5200 racks w/ 6 trays of 22 10K rpm 18G drives

Redo Logs:

- Create 9-disk RAID0 LUN on each T3 brick (64K blocksize).
- Turn write cache on for each LUN.
- Using Veritas, stripe 4 LUNs from 4 T3 bricks (128K interlace) and mirror on
  4 LUNs from the other 4 T3 bricks.

Data/indexes:

Goals:

1. Stripe every logical entity (table, index, temp) to as many
    of the 1320 disks as possible.

2. Keep the number of objects in a disk group as low as possible.

Limitations:

1. Striping volumes 1320 wide would only allow about 20 volumes to be created
    in VxVM without making a very large private region (20M holds ~25K objects)

What we did was to create 6 VxVM disk groups on the A5200 rotating through
the trays so each disk group got approximately the same number of disks from
each tray, with 220 disks per disk group.

For RAID0+1 volumes (all but temp), We striped across 110 disks from each
controller and mirrored to the other 110 disks. Every logical entity was
created on multiple volumes, e.g. the tablespace for LINEITEM was created on
84 volumes. Using round-robin assignment, these volumes rotated between the
6 disk groups. In this way the LINEITEM table was spread across all 1320 disks.

For RAID0 volumes (temp), we striped across all 220 disks from each disk
group. Again the logical entity (e.g. tablespace for temp) was
created on multiple volumes so that again it ended up striped across all 1320
disks.

Following is the volume manager description of typical volumes for each entity.

```
% vxprint -th l_1

Disk group: photon1

V  NAME         USETYPE      KSTATE   STATE    LENGTH
READPOL   PREFPLEX
PL NAME         VOLUME       KSTATE   STATE    LENGTH
LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH
[COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH
[COL/]OFF AM/NM    MODE

v  l_1          gen          ENABLED  ACTIVE
67133440 PREFER    l_1-02
pl l_1-01       l_1          ENABLED  ACTIVE
67133440 STRIPE    110/2048 RW
sd c10t100d0s2-01 l_1-01     c10t100d0s2 0     610304
0/0       c15t100d0 ENA
sd c11t32d0s2-01 l_1-01      c11t32d0s2 0      610304
1/0       c6t32d0  ENA
sd c12t64d0s2-01 l_1-01      c12t64d0s2 0      610304
2/0       c10t64d0 ENA
sd c13t0d0s2-01 l_1-01       c13t0d0s2 0       610304
3/0       c17t0d0  ENA
sd c14t64d0s2-01 l_1-01      c14t64d0s2 0      610304
4/0       c19t64d0 ENA
sd c15t100d0s2-01 l_1-01     c15t100d0s2 0     610304
5/0       c13t100d0 ENA
sd c16t32d0s2-01 l_1-01      c16t32d0s2 0      610304
6/0       c7t32d0  ENA
sd c17t64d0s2-01 l_1-01      c17t64d0s2 0      610304
7/0       c4t64d0  ENA
sd c18t100d0s2-01 l_1-01     c18t100d0s2 0     610304
8/0       c31t100d0 ENA
```

```
sd c19t32d0s2-01 l_1-01      c19t32d0s2 0      610304
9/0       c25t32d0 ENA
sd c20t64d0s2-01 l_1-01      c20t64d0s2 0      610304
10/0      c32t64d0 ENA
sd c21t100d0s2-01 l_1-01     c21t100d0s2 0     610304
11/0      c34t100d0 ENA
sd c22t0d0s2-01 l_1-01       c22t0d0s2 0       610304
12/0      c24t0d0  ENA
sd c23t32d0s2-01 l_1-01      c23t32d0s2 0      610304
13/0      c41t32d0 ENA
sd c24t0d0s2-01 l_1-01       c24t0d0s2 0       610304
14/0      c33t0d0  ENA
sd c25t0d0s2-01 l_1-01       c25t0d0s2 0       610304
15/0      c37t0d0  ENA
sd c26t0d0s2-01 l_1-01       c26t0d0s2 0       610304
16/0      c36t0d0  ENA
sd c27t100d0s2-01 l_1-01     c27t100d0s2 0     610304
17/0      c29t100d0 ENA
sd c28t0d0s2-01 l_1-01       c28t0d0s2 0       610304
18/0      c39t0d0  ENA
sd c29t100d0s2-01 l_1-01     c29t100d0s2 0     610304
19/0      c28t100d0 ENA
sd c30t100d0s2-01 l_1-01     c30t100d0s2 0     610304
20/0      c27t100d0 ENA
sd c31t64d0s2-01 l_1-01      c31t64d0s2 0      610304
21/0      c38t64d0 ENA
sd c32t32d0s2-01 l_1-01      c32t32d0s2 0      610304
22/0      c40t32d0 ENA
sd c33t64d0s2-01 l_1-01      c33t64d0s2 0      610304
23/0      c42t64d0 ENA
sd c34t32d0s2-01 l_1-01      c34t32d0s2 0      610304
24/0      c46t32d0 ENA
sd c35t64d0s2-01 l_1-01      c35t64d0s2 0      610304
25/0      c43t64d0 ENA
sd c36t0d0s2-01 l_1-01       c36t0d0s2 0       610304
26/0      c51t0d0  ENA
sd c37t64d0s2-01 l_1-01      c37t64d0s2 0      610304
27/0      c44t64d0 ENA
sd c38t100d0s2-01 l_1-01     c38t100d0s2 0     610304
28/0      c50t100d0 ENA
sd c39t0d0s2-01 l_1-01       c39t0d0s2 0       610304
29/0      c30t0d0  ENA
sd c40t32d0s2-01 l_1-01      c40t32d0s2 0      610304
30/0      c26t32d0 ENA
sd c41t32d0s2-01 l_1-01      c41t32d0s2 0      610304
31/0      c47t32d0 ENA
sd c42t64d0s2-01 l_1-01      c42t64d0s2 0      610304
32/0      c49t64d0 ENA
sd c43t100d0s2-01 l_1-01     c43t100d0s2 0     610304
33/0      c48t100d0 ENA
sd c44t100d0s2-01 l_1-01     c44t100d0s2 0     610304
34/0      c23t100d0 ENA
sd c45t0d0s2-01 l_1-01       c45t0d0s2 0       610304
35/0      c22t0d0  ENA
sd c46t0d0s2-01 l_1-01       c46t0d0s2 0       610304
36/0      c21t0d0  ENA
sd c47t32d0s2-01 l_1-01      c47t32d0s2 0      610304
37/0      c3t32d0  ENA
sd c48t32d0s2-01 l_1-01      c48t32d0s2 0      610304
38/0      c20t32d0 ENA
sd c49t64d0s2-01 l_1-01      c49t64d0s2 0      610304
39/0      c8t64d0  ENA
sd c10t68d0s2-01 l_1-01      c10t68d0s2 0      610304
40/0      c15t68d0 ENA
sd c13t18d0s2-01 l_1-01      c13t18d0s2 0      610304
41/0      c17t18d0 ENA
sd c15t68d0s2-01 l_1-01      c15t68d0s2 0      610304
42/0      c13t68d0 ENA
sd c18t68d0s2-01 l_1-01      c18t68d0s2 0      610304
43/0      c31t68d0 ENA
sd c21t68d0s2-01 l_1-01      c21t68d0s2 0      610304
44/0      c34t68d0 ENA
sd c22t18d0s2-01 l_1-01      c22t18d0s2 0      610304
45/0      c24t18d0 ENA
sd c24t18d0s2-01 l_1-01      c24t18d0s2 0      610304
46/0      c33t18d0 ENA
sd c25t18d0s2-01 l_1-01      c25t18d0s2 0      610304
47/0      c37t18d0 ENA
sd c26t18d0s2-01 l_1-01      c26t18d0s2 0      610304
48/0      c36t18d0 ENA
sd c27t68d0s2-01 l_1-01      c27t68d0s2 0      610304
49/0      c29t68d0 ENA
sd c28t18d0s2-01 l_1-01      c28t18d0s2 0      610304
50/0      c39t18d0 ENA
sd c29t68d0s2-01 l_1-01      c29t68d0s2 0      610304
51/0      c28t68d0 ENA
sd c30t68d0s2-01 l_1-01      c30t68d0s2 0      610304
52/0      c27t68d0 ENA
sd c36t18d0s2-01 l_1-01      c36t18d0s2 0      610304
53/0      c51t18d0 ENA
sd c38t68d0s2-01 l_1-01      c38t68d0s2 0      610304
54/0      c50t68d0 ENA
sd c39t18d0s2-01 l_1-01      c39t18d0s2 0      610304
55/0      c30t18d0 ENA
sd c43t68d0s2-01 l_1-01      c43t68d0s2 0      610304
56/0      c48t68d0 ENA
sd c44t68d0s2-01 l_1-01      c44t68d0s2 0      610304
57/0      c23t68d0 ENA
sd c45t18d0s2-01 l_1-01      c45t18d0s2 0      610304
58/0      c22t18d0 ENA
sd c46t18d0s2-01 l_1-01      c46t18d0s2 0      610304
59/0      c21t18d0 ENA
sd c10t101d0s2-01 l_1-01     c10t101d0s2 0     610304
60/0      c15t101d0 ENA
sd c11t33d0s2-01 l_1-01      c11t33d0s2 0      610304
61/0      c6t33d0  ENA
sd c12t65d0s2-01 l_1-01      c12t65d0s2 0      610304
62/0      c10t65d0 ENA
sd c13t100d0s2-01 l_1-01     c13t100d0s2 0     610304
63/0      c17t100d0 ENA
sd c14t65d0s2-01 l_1-01      c14t65d0s2 0      610304
64/0      c19t65d0 ENA
sd c15t101d0s2-01 l_1-01     c15t101d0s2 0     610304
65/0      c13t101d0 ENA
sd c16t33d0s2-01 l_1-01      c16t33d0s2 0      610304
66/0      c7t33d0  ENA
sd c17t65d0s2-01 l_1-01      c17t65d0s2 0      610304
67/0      c4t65d0  ENA
sd c18t101d0s2-01 l_1-01     c18t101d0s2 0     610304
68/0      c31t101d0 ENA
sd c19t33d0s2-01 l_1-01      c19t33d0s2 0      610304
69/0      c25t33d0 ENA
sd c20t65d0s2-01 l_1-01      c20t65d0s2 0      610304
70/0      c32t65d0 ENA
sd c21t101d0s2-01 l_1-01     c21t101d0s2 0     610304
71/0      c34t101d0 ENA
sd c22t100d0s2-01 l_1-01     c22t100d0s2 0     610304
72/0      c24t100d0 ENA
sd c23t33d0s2-01 l_1-01      c23t33d0s2 0      610304
73/0      c41t33d0 ENA
sd c24t100d0s2-01 l_1-01     c24t100d0s2 0     610304
74/0      c33t100d0 ENA
sd c25t100d0s2-01 l_1-01     c25t100d0s2 0     610304
75/0      c37t100d0 ENA
sd c26t100d0s2-01 l_1-01     c26t100d0s2 0     610304
76/0      c36t100d0 ENA
sd c27t101d0s2-01 l_1-01     c27t101d0s2 0     610304
77/0      c29t101d0 ENA
sd c28t100d0s2-01 l_1-01     c28t100d0s2 0     610304
78/0      c39t100d0 ENA
sd c29t101d0s2-01 l_1-01     c29t101d0s2 0     610304
79/0      c28t101d0 ENA
sd c30t101d0s2-01 l_1-01     c30t101d0s2 0     610304
80/0      c27t101d0 ENA
sd c31t65d0s2-01 l_1-01      c31t65d0s2 0      610304
81/0      c38t65d0 ENA
sd c32t33d0s2-01 l_1-01      c32t33d0s2 0      610304
82/0      c40t33d0 ENA
sd c33t65d0s2-01 l_1-01      c33t65d0s2 0      610304
83/0      c42t65d0 ENA
sd c34t33d0s2-01 l_1-01      c34t33d0s2 0      610304
84/0      c46t33d0 ENA
sd c35t65d0s2-01 l_1-01      c35t65d0s2 0      610304
85/0      c43t65d0 ENA
sd c36t100d0s2-01 l_1-01     c36t100d0s2 0     610304
86/0      c51t100d0 ENA
sd c37t65d0s2-01 l_1-01      c37t65d0s2 0      610304
87/0      c44t65d0 ENA
sd c38t101d0s2-01 l_1-01     c38t101d0s2 0     610304
88/0      c50t101d0 ENA
sd c39t100d0s2-01 l_1-01     c39t100d0s2 0     610304
89/0      c30t100d0 ENA
sd c40t33d0s2-01 l_1-01      c40t33d0s2 0      610304
90/0      c26t33d0 ENA
sd c41t33d0s2-01 l_1-01      c41t33d0s2 0      610304
91/0      c47t33d0 ENA
sd c42t65d0s2-01 l_1-01      c42t65d0s2 0      610304
92/0      c49t65d0 ENA
sd c43t101d0s2-01 l_1-01     c43t101d0s2 0     610304
93/0      c48t101d0 ENA
sd c44t101d0s2-01 l_1-01     c44t101d0s2 0     610304
94/0      c23t101d0 ENA
sd c45t100d0s2-01 l_1-01     c45t100d0s2 0     610304
95/0      c22t100d0 ENA
sd c46t100d0s2-01 l_1-01     c46t100d0s2 0     610304
96/0      c21t100d0 ENA
sd c47t33d0s2-01 l_1-01      c47t33d0s2 0      610304
97/0      c3t33d0  ENA
sd c48t33d0s2-01 l_1-01      c48t33d0s2 0      610304
98/0      c20t33d0 ENA
```

```
sd c49t65d0s2-01 l_1-01      c49t65d0s2 0      610304
99/0      c8t65d0  ENA
sd c10t69d0s2-01 l_1-01      c10t69d0s2 0      610304
100/0     c15t69d0 ENA
sd c13t19d0s2-01 l_1-01      c13t19d0s2 0      610304
101/0     c17t19d0 ENA
sd c15t69d0s2-01 l_1-01      c15t69d0s2 0      610304
102/0     c13t69d0 ENA
sd c18t69d0s2-01 l_1-01      c18t69d0s2 0      610304
103/0     c31t69d0 ENA
sd c21t69d0s2-01 l_1-01      c21t69d0s2 0      610304
104/0     c34t69d0 ENA
sd c22t19d0s2-01 l_1-01      c22t19d0s2 0      610304
105/0     c24t19d0 ENA
sd c24t19d0s2-01 l_1-01      c24t19d0s2 0      610304
106/0     c33t19d0 ENA
sd c25t19d0s2-01 l_1-01      c25t19d0s2 0      610304
107/0     c37t19d0 ENA
sd c26t19d0s2-01 l_1-01      c26t19d0s2 0      610304
108/0     c36t19d0 ENA
sd c27t69d0s2-01 l_1-01      c27t69d0s2 0      610304
109/0     c29t69d0 ENA
pl l_1-02       l_1          ENABLED  ACTIVE
67133440 STRIPE    110/2048 RW
sd c28t19d0s2-01 l_1-02      c28t19d0s2 0      610304
0/0       c39t19d0 ENA
sd c29t69d0s2-01 l_1-02      c29t69d0s2 0      610304
1/0       c28t69d0 ENA
sd c30t69d0s2-01 l_1-02      c30t69d0s2 0      610304
2/0       c27t69d0 ENA
sd c36t19d0s2-01 l_1-02      c36t19d0s2 0      610304
3/0       c51t19d0 ENA
sd c38t69d0s2-01 l_1-02      c38t69d0s2 0      610304
4/0       c50t69d0 ENA
sd c39t19d0s2-01 l_1-02      c39t19d0s2 0      610304
5/0       c30t19d0 ENA
sd c43t69d0s2-01 l_1-02      c43t69d0s2 0      610304
6/0       c48t69d0 ENA
sd c44t69d0s2-01 l_1-02      c44t69d0s2 0      610304
7/0       c23t69d0 ENA
sd c45t19d0s2-01 l_1-02      c45t19d0s2 0      610304
8/0       c22t19d0 ENA
sd c46t19d0s2-01 l_1-02      c46t19d0s2 0      610304
9/0       c21t19d0 ENA
sd c10t102d0s2-01 l_1-02     c10t102d0s2 0     610304
10/0      c15t102d0 ENA
sd c11t34d0s2-01 l_1-02      c11t34d0s2 0      610304
11/0      c6t34d0  ENA
sd c12t66d0s2-01 l_1-02      c12t66d0s2 0      610304
12/0      c10t66d0 ENA
sd c13t101d0s2-01 l_1-02     c13t101d0s2 0     610304
13/0      c17t101d0 ENA
sd c14t66d0s2-01 l_1-02      c14t66d0s2 0      610304
14/0      c19t66d0 ENA
sd c15t102d0s2-01 l_1-02     c15t102d0s2 0     610304
15/0      c13t102d0 ENA
sd c16t34d0s2-01 l_1-02      c16t34d0s2 0      610304
16/0      c7t34d0  ENA
sd c17t66d0s2-01 l_1-02      c17t66d0s2 0      610304
17/0      c4t66d0  ENA
sd c18t102d0s2-01 l_1-02     c18t102d0s2 0     610304
18/0      c31t102d0 ENA
sd c19t34d0s2-01 l_1-02      c19t34d0s2 0      610304
19/0      c25t34d0 ENA
sd c20t66d0s2-01 l_1-02      c20t66d0s2 0      610304
20/0      c32t66d0 ENA
sd c21t102d0s2-01 l_1-02     c21t102d0s2 0     610304
21/0      c34t102d0 ENA
sd c22t101d0s2-01 l_1-02     c22t101d0s2 0     610304
22/0      c24t101d0 ENA
sd c23t34d0s2-01 l_1-02      c23t34d0s2 0      610304
23/0      c41t34d0 ENA
sd c24t101d0s2-01 l_1-02     c24t101d0s2 0     610304
24/0      c33t101d0 ENA
sd c25t101d0s2-01 l_1-02     c25t101d0s2 0     610304
25/0      c37t101d0 ENA
sd c26t101d0s2-01 l_1-02     c26t101d0s2 0     610304
26/0      c36t101d0 ENA
sd c27t102d0s2-01 l_1-02     c27t102d0s2 0     610304
27/0      c29t102d0 ENA
sd c28t101d0s2-01 l_1-02     c28t101d0s2 0     610304
28/0      c39t101d0 ENA
sd c29t102d0s2-01 l_1-02     c29t102d0s2 0     610304
29/0      c28t102d0 ENA
sd c30t102d0s2-01 l_1-02     c30t102d0s2 0     610304
30/0      c27t102d0 ENA
sd c31t66d0s2-01 l_1-02      c31t66d0s2 0      610304
31/0      c38t66d0 ENA
sd c32t34d0s2-01 l_1-02      c32t34d0s2 0      610304
32/0      c40t34d0 ENA
sd c33t66d0s2-01 l_1-02      c33t66d0s2 0      610304
33/0      c42t66d0 ENA
sd c34t34d0s2-01 l_1-02      c34t34d0s2 0      610304
34/0      c46t34d0 ENA
sd c35t66d0s2-01 l_1-02      c35t66d0s2 0      610304
35/0      c43t66d0 ENA
sd c36t101d0s2-01 l_1-02     c36t101d0s2 0     610304
36/0      c51t101d0 ENA
sd c37t66d0s2-01 l_1-02      c37t66d0s2 0      610304
37/0      c44t66d0 ENA
sd c38t102d0s2-01 l_1-02     c38t102d0s2 0     610304
38/0      c50t102d0 ENA
sd c39t101d0s2-01 l_1-02     c39t101d0s2 0     610304
39/0      c30t101d0 ENA
sd c40t34d0s2-01 l_1-02      c40t34d0s2 0      610304
40/0      c26t34d0 ENA
sd c41t34d0s2-01 l_1-02      c41t34d0s2 0      610304
41/0      c47t34d0 ENA
sd c42t66d0s2-01 l_1-02      c42t66d0s2 0      610304
42/0      c49t66d0 ENA
sd c43t102d0s2-01 l_1-02     c43t102d0s2 0     610304
43/0      c48t102d0 ENA
sd c44t102d0s2-01 l_1-02     c44t102d0s2 0     610304
44/0      c23t102d0 ENA
sd c45t101d0s2-01 l_1-02     c45t101d0s2 0     610304
45/0      c22t101d0 ENA
sd c46t101d0s2-01 l_1-02     c46t101d0s2 0     610304
46/0      c21t101d0 ENA
sd c47t34d0s2-01 l_1-02      c47t34d0s2 0      610304
47/0      c3t34d0  ENA
sd c48t34d0s2-01 l_1-02      c48t34d0s2 0      610304
48/0      c20t34d0 ENA
sd c49t66d0s2-01 l_1-02      c49t66d0s2 0      610304
49/0      c8t66d0  ENA
sd c10t70d0s2-01 l_1-02      c10t70d0s2 0      610304
50/0      c15t70d0 ENA
sd c13t1d0s2-01 l_1-02       c13t1d0s2 0       610304
51/0      c17t1d0  ENA
sd c15t70d0s2-01 l_1-02      c15t70d0s2 0      610304
52/0      c13t70d0 ENA
sd c18t70d0s2-01 l_1-02      c18t70d0s2 0      610304
53/0      c31t70d0 ENA
sd c21t70d0s2-01 l_1-02      c21t70d0s2 0      610304
54/0      c34t70d0 ENA
sd c22t1d0s2-01 l_1-02       c22t1d0s2 0       610304
55/0      c24t1d0  ENA
sd c24t1d0s2-01 l_1-02       c24t1d0s2 0       610304
56/0      c33t1d0  ENA
sd c25t1d0s2-01 l_1-02       c25t1d0s2 0       610304
57/0      c37t1d0  ENA
sd c26t1d0s2-01 l_1-02       c26t1d0s2 0       610304
58/0      c36t1d0  ENA
sd c27t70d0s2-01 l_1-02      c27t70d0s2 0      610304
59/0      c29t70d0 ENA
sd c28t1d0s2-01 l_1-02       c28t1d0s2 0       610304
60/0      c39t1d0  ENA
sd c29t70d0s2-01 l_1-02      c29t70d0s2 0      610304
61/0      c28t70d0 ENA
sd c30t70d0s2-01 l_1-02      c30t70d0s2 0      610304
62/0      c27t70d0 ENA
sd c36t1d0s2-01 l_1-02       c36t1d0s2 0       610304
63/0      c51t1d0  ENA
sd c38t70d0s2-01 l_1-02      c38t70d0s2 0      610304
64/0      c50t70d0 ENA
sd c39t1d0s2-01 l_1-02       c39t1d0s2 0       610304
65/0      c30t1d0  ENA
sd c43t70d0s2-01 l_1-02      c43t70d0s2 0      610304
66/0      c48t70d0 ENA
sd c44t70d0s2-01 l_1-02      c44t70d0s2 0      610304
67/0      c23t70d0 ENA
sd c45t1d0s2-01 l_1-02       c45t1d0s2 0       610304
68/0      c22t1d0  ENA
sd c46t1d0s2-01 l_1-02       c46t1d0s2 0       610304
69/0      c21t1d0  ENA
sd c10t103d0s2-01 l_1-02     c10t103d0s2 0     610304
70/0      c15t103d0 ENA
sd c11t35d0s2-01 l_1-02      c11t35d0s2 0      610304
71/0      c6t35d0  ENA
sd c12t67d0s2-01 l_1-02      c12t67d0s2 0      610304
72/0      c10t67d0 ENA
sd c13t102d0s2-01 l_1-02     c13t102d0s2 0     610304
73/0      c17t102d0 ENA
sd c14t67d0s2-01 l_1-02      c14t67d0s2 0      610304
74/0      c19t67d0 ENA
sd c15t103d0s2-01 l_1-02     c15t103d0s2 0     610304
75/0      c13t103d0 ENA
sd c16t35d0s2-01 l_1-02      c16t35d0s2 0      610304
76/0      c7t35d0  ENA
sd c17t67d0s2-01 l_1-02      c17t67d0s2 0      610304
77/0      c4t67d0  ENA
```

```
sd c18t103d0s2-01 l_1-02     c18t103d0s2 0     610304
78/0      c31t103d0 ENA
sd c19t35d0s2-01 l_1-02      c19t35d0s2 0      610304
79/0      c25t35d0 ENA
sd c20t67d0s2-01 l_1-02      c20t67d0s2 0      610304
80/0      c32t67d0 ENA
sd c21t103d0s2-01 l_1-02     c21t103d0s2 0     610304
81/0      c34t103d0 ENA
sd c22t102d0s2-01 l_1-02     c22t102d0s2 0     610304
82/0      c24t102d0 ENA
sd c23t35d0s2-01 l_1-02      c23t35d0s2 0      610304
83/0      c41t35d0 ENA
sd c24t102d0s2-01 l_1-02     c24t102d0s2 0     610304
84/0      c33t102d0 ENA
sd c25t102d0s2-01 l_1-02     c25t102d0s2 0     610304
85/0      c37t102d0 ENA
sd c26t102d0s2-01 l_1-02     c26t102d0s2 0     610304
86/0      c36t102d0 ENA
sd c27t103d0s2-01 l_1-02     c27t103d0s2 0     610304
87/0      c29t103d0 ENA
sd c28t102d0s2-01 l_1-02     c28t102d0s2 0     610304
88/0      c39t102d0 ENA
sd c29t103d0s2-01 l_1-02     c29t103d0s2 0     610304
89/0      c28t103d0 ENA
sd c30t103d0s2-01 l_1-02     c30t103d0s2 0     610304
90/0      c27t103d0 ENA
sd c31t67d0s2-01 l_1-02      c31t67d0s2 0      610304
91/0      c38t67d0 ENA
sd c32t35d0s2-01 l_1-02      c32t35d0s2 0      610304
92/0      c40t35d0 ENA
sd c33t67d0s2-01 l_1-02      c33t67d0s2 0      610304
93/0      c42t67d0 ENA
sd c34t35d0s2-01 l_1-02      c34t35d0s2 0      610304
94/0      c46t35d0 ENA
sd c35t67d0s2-01 l_1-02      c35t67d0s2 0      610304
95/0      c43t67d0 ENA
sd c36t102d0s2-01 l_1-02     c36t102d0s2 0     610304
96/0      c51t102d0 ENA
sd c37t67d0s2-01 l_1-02      c37t67d0s2 0      610304
97/0      c44t67d0 ENA
sd c38t103d0s2-01 l_1-02     c38t103d0s2 0     610304
98/0      c50t103d0 ENA
sd c39t102d0s2-01 l_1-02     c39t102d0s2 0     610304
99/0      c30t102d0 ENA
sd c40t35d0s2-01 l_1-02      c40t35d0s2 0      610304
100/0     c26t35d0 ENA
sd c41t35d0s2-01 l_1-02      c41t35d0s2 0      610304
101/0     c47t35d0 ENA
sd c42t67d0s2-01 l_1-02      c42t67d0s2 0      610304
102/0     c49t67d0 ENA
sd c43t103d0s2-01 l_1-02     c43t103d0s2 0     610304
103/0     c48t103d0 ENA
sd c44t103d0s2-01 l_1-02     c44t103d0s2 0     610304
104/0     c23t103d0 ENA
sd c45t102d0s2-01 l_1-02     c45t102d0s2 0     610304
105/0     c22t102d0 ENA
sd c46t102d0s2-01 l_1-02     c46t102d0s2 0     610304
106/0     c21t102d0 ENA
sd c47t35d0s2-01 l_1-02      c47t35d0s2 0      610304
107/0     c3t35d0  ENA
sd c48t35d0s2-01 l_1-02      c48t35d0s2 0      610304
108/0     c20t35d0 ENA
sd c49t67d0s2-01 l_1-02      c49t67d0s2 0      610304
109/0     c8t67d0  ENA



% vxprint -th temp_6

Disk group: photon2

V  NAME         USETYPE      KSTATE   STATE    LENGTH
READPOL   PREFPLEX
PL NAME         VOLUME       KSTATE   STATE    LENGTH
LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH
[COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH
[COL/]OFF AM/NM    MODE

v  temp_6       gen          ENABLED  ACTIVE
67133440 ROUND     -
pl temp_6-01    temp_6       ENABLED  ACTIVE
67133440 STRIPE    220/1024 RW
sd c15t72d0s2-49 temp_6-01   c15t72d0s2 19290112
305152 0/0       c13t72d0 ENA
sd c18t72d0s2-49 temp_6-01   c18t72d0s2 19290112
305152 1/0       c31t72d0 ENA
sd c21t72d0s2-49 temp_6-01   c21t72d0s2 19290112
305152 2/0       c34t72d0 ENA
sd c22t21d0s2-49 temp_6-01   c22t21d0s2 19290112
305152 3/0       c24t21d0 ENA
sd c24t21d0s2-49 temp_6-01   c24t21d0s2 19290112
305152 4/0       c33t21d0 ENA
sd c25t21d0s2-49 temp_6-01   c25t21d0s2 19290112
305152 5/0       c37t21d0 ENA
sd c26t21d0s2-49 temp_6-01   c26t21d0s2 19290112
305152 6/0       c36t21d0 ENA
sd c27t72d0s2-49 temp_6-01   c27t72d0s2 19290112
305152 7/0       c29t72d0 ENA
sd c28t21d0s2-49 temp_6-01   c28t21d0s2 19290112
305152 8/0       c39t21d0 ENA
sd c29t72d0s2-49 temp_6-01   c29t72d0s2 19290112
305152 9/0       c28t72d0 ENA
sd c30t72d0s2-49 temp_6-01   c30t72d0s2 19290112
305152 10/0      c27t72d0 ENA
sd c36t21d0s2-49 temp_6-01   c36t21d0s2 19290112
305152 11/0      c51t21d0 ENA
sd c38t72d0s2-49 temp_6-01   c38t72d0s2 19290112
305152 12/0      c50t72d0 ENA
sd c39t21d0s2-49 temp_6-01   c39t21d0s2 19290112
305152 13/0      c30t21d0 ENA
sd c43t72d0s2-49 temp_6-01   c43t72d0s2 19290112
305152 14/0      c48t72d0 ENA
sd c44t72d0s2-49 temp_6-01   c44t72d0s2 19290112
305152 15/0      c23t72d0 ENA
sd c45t21d0s2-49 temp_6-01   c45t21d0s2 19290112
305152 16/0      c22t21d0 ENA
sd c46t21d0s2-49 temp_6-01   c46t21d0s2 19290112
305152 17/0      c21t21d0 ENA
sd c10t105d0s2-49 temp_6-01  c10t105d0s2 19290112
305152 18/0     c15t105d0 ENA
sd c11t37d0s2-49 temp_6-01   c11t37d0s2 19290112
305152 19/0      c6t37d0  ENA
sd c12t69d0s2-49 temp_6-01   c12t69d0s2 19290112
305152 20/0      c10t69d0 ENA
sd c13t104d0s2-49 temp_6-01  c13t104d0s2 19290112
305152 21/0     c17t104d0 ENA
sd c14t69d0s2-49 temp_6-01   c14t69d0s2 19290112
305152 22/0      c19t69d0 ENA
sd c15t105d0s2-49 temp_6-01  c15t105d0s2 19290112
305152 23/0     c13t105d0 ENA
sd c16t37d0s2-49 temp_6-01   c16t37d0s2 19290112
305152 24/0      c7t37d0  ENA
sd c17t69d0s2-49 temp_6-01   c17t69d0s2 19290112
305152 25/0      c4t69d0  ENA
sd c18t105d0s2-49 temp_6-01  c18t105d0s2 19290112
305152 26/0     c31t105d0 ENA
sd c19t37d0s2-49 temp_6-01   c19t37d0s2 19290112
305152 27/0      c25t37d0 ENA
sd c20t69d0s2-49 temp_6-01   c20t69d0s2 19290112
305152 28/0      c32t69d0 ENA
sd c21t105d0s2-49 temp_6-01  c21t105d0s2 19290112
305152 29/0     c34t105d0 ENA
sd c22t104d0s2-49 temp_6-01  c22t104d0s2 19290112
305152 30/0     c24t104d0 ENA
sd c23t37d0s2-49 temp_6-01   c23t37d0s2 19290112
305152 31/0      c41t37d0 ENA
sd c24t104d0s2-49 temp_6-01  c24t104d0s2 19290112
305152 32/0     c33t104d0 ENA
sd c25t104d0s2-49 temp_6-01  c25t104d0s2 19290112
305152 33/0     c37t104d0 ENA
sd c26t104d0s2-49 temp_6-01  c26t104d0s2 19290112
305152 34/0     c36t104d0 ENA
sd c27t105d0s2-49 temp_6-01  c27t105d0s2 19290112
305152 35/0     c29t105d0 ENA
sd c28t104d0s2-49 temp_6-01  c28t104d0s2 19290112
305152 36/0     c39t104d0 ENA
sd c29t105d0s2-49 temp_6-01  c29t105d0s2 19290112
305152 37/0     c28t105d0 ENA
sd c30t105d0s2-49 temp_6-01  c30t105d0s2 19290112
305152 38/0     c27t105d0 ENA
sd c31t69d0s2-49 temp_6-01   c31t69d0s2 19290112
305152 39/0      c38t69d0 ENA
sd c32t37d0s2-49 temp_6-01   c32t37d0s2 19290112
305152 40/0      c40t37d0 ENA
sd c33t69d0s2-49 temp_6-01   c33t69d0s2 19290112
305152 41/0      c42t69d0 ENA
sd c34t37d0s2-49 temp_6-01   c34t37d0s2 19290112
305152 42/0      c46t37d0 ENA
sd c35t69d0s2-49 temp_6-01   c35t69d0s2 19290112
305152 43/0      c43t69d0 ENA
sd c36t104d0s2-49 temp_6-01  c36t104d0s2 19290112
305152 44/0     c51t104d0 ENA
sd c37t69d0s2-49 temp_6-01   c37t69d0s2 19290112
305152 45/0      c44t69d0 ENA
sd c38t105d0s2-49 temp_6-01  c38t105d0s2 19290112
305152 46/0     c50t105d0 ENA
sd c39t104d0s2-49 temp_6-01  c39t104d0s2 19290112
305152 47/0     c30t104d0 ENA
```

```
sd c40t37d0s2-49 temp_6-01   c40t37d0s2 19290112
305152 48/0      c26t37d0 ENA
sd c41t37d0s2-49 temp_6-01   c41t37d0s2 19290112
305152 49/0      c47t37d0 ENA
sd c42t69d0s2-49 temp_6-01   c42t69d0s2 19290112
305152 50/0      c49t69d0 ENA
sd c43t105d0s2-49 temp_6-01  c43t105d0s2 19290112
305152 51/0      c48t105d0 ENA
sd c44t105d0s2-49 temp_6-01  c44t105d0s2 19290112
305152 52/0      c23t105d0 ENA
sd c45t104d0s2-49 temp_6-01  c45t104d0s2 19290112
305152 53/0      c22t104d0 ENA
sd c46t104d0s2-49 temp_6-01  c46t104d0s2 19290112
305152 54/0      c21t104d0 ENA
sd c47t37d0s2-49 temp_6-01   c47t37d0s2 19290112
305152 55/0      c3t37d0  ENA
sd c48t37d0s2-49 temp_6-01   c48t37d0s2 19290112
305152 56/0      c20t37d0 ENA
sd c49t69d0s2-49 temp_6-01   c49t69d0s2 19290112
305152 57/0      c8t69d0  ENA
sd c10t73d0s2-49 temp_6-01   c10t73d0s2 19290112
305152 58/0      c15t73d0 ENA
sd c13t22d0s2-49 temp_6-01   c13t22d0s2 19290112
305152 59/0      c17t22d0 ENA
sd c15t73d0s2-49 temp_6-01   c15t73d0s2 19290112
305152 60/0      c13t73d0 ENA
sd c18t73d0s2-49 temp_6-01   c18t73d0s2 19290112
305152 61/0      c31t73d0 ENA
sd c21t73d0s2-49 temp_6-01   c21t73d0s2 19290112
305152 62/0      c34t73d0 ENA
sd c22t22d0s2-49 temp_6-01   c22t22d0s2 19290112
305152 63/0      c24t22d0 ENA
sd c24t22d0s2-49 temp_6-01   c24t22d0s2 19290112
305152 64/0      c33t22d0 ENA
sd c25t22d0s2-49 temp_6-01   c25t22d0s2 19290112
305152 65/0      c37t22d0 ENA
sd c26t22d0s2-49 temp_6-01   c26t22d0s2 19290112
305152 66/0      c36t22d0 ENA
sd c27t73d0s2-49 temp_6-01   c27t73d0s2 19290112
305152 67/0      c29t73d0 ENA
sd c28t22d0s2-49 temp_6-01   c28t22d0s2 19290112
305152 68/0      c39t22d0 ENA
sd c29t73d0s2-49 temp_6-01   c29t73d0s2 19290112
305152 69/0      c28t73d0 ENA
sd c30t73d0s2-49 temp_6-01   c30t73d0s2 19290112
305152 70/0      c27t73d0 ENA
sd c36t22d0s2-49 temp_6-01   c36t22d0s2 19290112
305152 71/0      c51t22d0 ENA
sd c38t73d0s2-49 temp_6-01   c38t73d0s2 19290112
305152 72/0      c50t73d0 ENA
sd c39t22d0s2-49 temp_6-01   c39t22d0s2 19290112
305152 73/0      c30t22d0 ENA
sd c43t73d0s2-49 temp_6-01   c43t73d0s2 19290112
305152 74/0      c48t73d0 ENA
sd c44t73d0s2-49 temp_6-01   c44t73d0s2 19290112
305152 75/0      c23t73d0 ENA
sd c45t22d0s2-49 temp_6-01   c45t22d0s2 19290112
305152 76/0      c22t22d0 ENA
sd c46t22d0s2-49 temp_6-01   c46t22d0s2 19290112
305152 77/0      c21t22d0 ENA
sd c10t106d0s2-49 temp_6-01  c10t106d0s2 19290112
305152 78/0      c15t106d0 ENA
sd c11t38d0s2-49 temp_6-01   c11t38d0s2 19290112
305152 79/0      c6t38d0  ENA
sd c12t70d0s2-49 temp_6-01   c12t70d0s2 19290112
305152 80/0      c10t70d0 ENA
sd c13t105d0s2-49 temp_6-01  c13t105d0s2 19290112
305152 81/0      c17t105d0 ENA
sd c14t70d0s2-49 temp_6-01   c14t70d0s2 19290112
305152 82/0      c19t70d0 ENA
sd c15t106d0s2-49 temp_6-01  c15t106d0s2 19290112
305152 83/0      c13t106d0 ENA
sd c16t38d0s2-49 temp_6-01   c16t38d0s2 19290112
305152 84/0      c7t38d0  ENA
sd c17t70d0s2-49 temp_6-01   c17t70d0s2 19290112
305152 85/0      c4t70d0  ENA
sd c18t106d0s2-49 temp_6-01  c18t106d0s2 19290112
305152 86/0      c31t106d0 ENA
sd c19t38d0s2-49 temp_6-01   c19t38d0s2 19290112
305152 87/0      c25t38d0 ENA
sd c20t70d0s2-49 temp_6-01   c20t70d0s2 19290112
305152 88/0      c32t70d0 ENA
sd c21t106d0s2-49 temp_6-01  c21t106d0s2 19290112
305152 89/0      c34t106d0 ENA
sd c22t105d0s2-49 temp_6-01  c22t105d0s2 19290112
305152 90/0      c24t105d0 ENA
sd c23t38d0s2-49 temp_6-01   c23t38d0s2 19290112
305152 91/0      c41t38d0 ENA
sd c24t105d0s2-49 temp_6-01  c24t105d0s2 19290112
305152 92/0      c33t105d0 ENA
sd c25t105d0s2-49 temp_6-01  c25t105d0s2 19290112
305152 93/0      c37t105d0 ENA
sd c26t105d0s2-49 temp_6-01  c26t105d0s2 19290112
305152 94/0      c36t105d0 ENA
sd c27t106d0s2-49 temp_6-01  c27t106d0s2 19290112
305152 95/0      c29t106d0 ENA
sd c28t105d0s2-49 temp_6-01  c28t105d0s2 19290112
305152 96/0      c39t105d0 ENA
sd c29t106d0s2-49 temp_6-01  c29t106d0s2 19290112
305152 97/0      c28t106d0 ENA
sd c30t106d0s2-49 temp_6-01  c30t106d0s2 19290112
305152 98/0      c27t106d0 ENA
sd c31t70d0s2-49 temp_6-01   c31t70d0s2 19290112
305152 99/0      c38t70d0 ENA
sd c32t38d0s2-49 temp_6-01   c32t38d0s2 19290112
305152 100/0     c40t38d0 ENA
sd c33t70d0s2-49 temp_6-01   c33t70d0s2 19290112
305152 101/0     c42t70d0 ENA
sd c34t38d0s2-49 temp_6-01   c34t38d0s2 19290112
305152 102/0     c46t38d0 ENA
sd c35t70d0s2-49 temp_6-01   c35t70d0s2 19290112
305152 103/0     c43t70d0 ENA
sd c36t105d0s2-49 temp_6-01  c36t105d0s2 19290112
305152 104/0     c51t105d0 ENA
sd c37t70d0s2-49 temp_6-01   c37t70d0s2 19290112
305152 105/0     c44t70d0 ENA
sd c38t106d0s2-49 temp_6-01  c38t106d0s2 19290112
305152 106/0     c50t106d0 ENA
sd c39t105d0s2-49 temp_6-01  c39t105d0s2 19290112
305152 107/0     c30t105d0 ENA
sd c40t38d0s2-49 temp_6-01   c40t38d0s2 19290112
305152 108/0     c26t38d0 ENA
sd c41t38d0s2-49 temp_6-01   c41t38d0s2 19290112
305152 109/0     c47t38d0 ENA
sd c42t70d0s2-49 temp_6-01   c42t70d0s2 19290112
305152 110/0     c49t70d0 ENA
sd c43t106d0s2-49 temp_6-01  c43t106d0s2 19290112
305152 111/0     c48t106d0 ENA
sd c44t106d0s2-49 temp_6-01  c44t106d0s2 19290112
305152 112/0     c23t106d0 ENA
sd c45t105d0s2-49 temp_6-01  c45t105d0s2 19290112
305152 113/0     c22t105d0 ENA
sd c46t105d0s2-49 temp_6-01  c46t105d0s2 19290112
305152 114/0     c21t105d0 ENA
sd c47t38d0s2-49 temp_6-01   c47t38d0s2 19290112
305152 115/0     c3t38d0  ENA
sd c48t38d0s2-49 temp_6-01   c48t38d0s2 19290112
305152 116/0     c20t38d0 ENA
sd c49t70d0s2-49 temp_6-01   c49t70d0s2 19290112
305152 117/0     c8t70d0  ENA
sd c10t74d0s2-49 temp_6-01   c10t74d0s2 19290112
305152 118/0     c15t74d0 ENA
sd c13t23d0s2-49 temp_6-01   c13t23d0s2 19290112
305152 119/0     c17t23d0 ENA
sd c15t74d0s2-49 temp_6-01   c15t74d0s2 19290112
305152 120/0     c13t74d0 ENA
sd c18t74d0s2-49 temp_6-01   c18t74d0s2 19290112
305152 121/0     c31t74d0 ENA
sd c21t74d0s2-49 temp_6-01   c21t74d0s2 19290112
305152 122/0     c34t74d0 ENA
sd c22t23d0s2-49 temp_6-01   c22t23d0s2 19290112
305152 123/0     c24t23d0 ENA
sd c24t23d0s2-49 temp_6-01   c24t23d0s2 19290112
305152 124/0     c33t23d0 ENA
sd c25t23d0s2-49 temp_6-01   c25t23d0s2 19290112
305152 125/0     c37t23d0 ENA
sd c26t23d0s2-49 temp_6-01   c26t23d0s2 19290112
305152 126/0     c36t23d0 ENA
sd c27t74d0s2-49 temp_6-01   c27t74d0s2 19290112
305152 127/0     c29t74d0 ENA
sd c28t23d0s2-49 temp_6-01   c28t23d0s2 19290112
305152 128/0     c39t23d0 ENA
sd c29t74d0s2-49 temp_6-01   c29t74d0s2 19290112
305152 129/0     c28t74d0 ENA
sd c30t74d0s2-49 temp_6-01   c30t74d0s2 19290112
305152 130/0     c27t74d0 ENA
sd c36t23d0s2-49 temp_6-01   c36t23d0s2 19290112
305152 131/0     c51t23d0 ENA
sd c38t74d0s2-49 temp_6-01   c38t74d0s2 19290112
305152 132/0     c50t74d0 ENA
sd c39t23d0s2-49 temp_6-01   c39t23d0s2 19290112
305152 133/0     c30t23d0 ENA
sd c43t74d0s2-49 temp_6-01   c43t74d0s2 19290112
305152 134/0     c48t74d0 ENA
sd c44t74d0s2-49 temp_6-01   c44t74d0s2 19290112
305152 135/0     c23t74d0 ENA
sd c45t23d0s2-49 temp_6-01   c45t23d0s2 19290112
305152 136/0     c22t23d0 ENA
sd c46t23d0s2-49 temp_6-01   c46t23d0s2 19290112
305152 137/0     c21t23d0 ENA
```

```
sd c10t112d0s2-49 temp_6-01  c10t112d0s2 19290112
305152 138/0    c15t112d0 ENA
sd c11t39d0s2-49 temp_6-01   c11t39d0s2 19290112
305152 139/0     c6t39d0  ENA
sd c12t71d0s2-49 temp_6-01   c12t71d0s2 19290112
305152 140/0     c10t71d0 ENA
sd c13t106d0s2-49 temp_6-01  c13t106d0s2 19290112
305152 141/0    c17t106d0 ENA
sd c14t71d0s2-49 temp_6-01   c14t71d0s2 19290112
305152 142/0     c19t71d0 ENA
sd c15t112d0s2-49 temp_6-01  c15t112d0s2 19290112
305152 143/0    c13t112d0 ENA
sd c16t39d0s2-49 temp_6-01   c16t39d0s2 19290112
305152 144/0     c7t39d0  ENA
sd c17t71d0s2-49 temp_6-01   c17t71d0s2 19290112
305152 145/0     c4t71d0  ENA
sd c18t112d0s2-49 temp_6-01  c18t112d0s2 19290112
305152 146/0    c31t112d0 ENA
sd c19t39d0s2-49 temp_6-01   c19t39d0s2 19290112
305152 147/0     c25t39d0 ENA
sd c20t71d0s2-49 temp_6-01   c20t71d0s2 19290112
305152 148/0     c32t71d0 ENA
sd c21t112d0s2-49 temp_6-01  c21t112d0s2 19290112
305152 149/0    c34t112d0 ENA
sd c22t106d0s2-49 temp_6-01  c22t106d0s2 19290112
305152 150/0    c24t106d0 ENA
sd c23t39d0s2-49 temp_6-01   c23t39d0s2 19290112
305152 151/0     c41t39d0 ENA
sd c24t106d0s2-49 temp_6-01  c24t106d0s2 19290112
305152 152/0    c33t106d0 ENA
sd c25t106d0s2-49 temp_6-01  c25t106d0s2 19290112
305152 153/0    c37t106d0 ENA
sd c26t106d0s2-49 temp_6-01  c26t106d0s2 19290112
305152 154/0    c36t106d0 ENA
sd c27t112d0s2-49 temp_6-01  c27t112d0s2 19290112
305152 155/0    c29t112d0 ENA
sd c28t106d0s2-49 temp_6-01  c28t106d0s2 19290112
305152 156/0    c39t106d0 ENA
sd c29t112d0s2-49 temp_6-01  c29t112d0s2 19290112
305152 157/0    c28t112d0 ENA
sd c10t71d0s2-49 temp_6-01   c10t71d0s2 19290112
305152 158/0     c15t71d0 ENA
sd c13t20d0s2-49 temp_6-01   c13t20d0s2 19290112
305152 159/0     c17t20d0 ENA
sd c15t71d0s2-49 temp_6-01   c15t71d0s2 19290112
305152 160/0     c13t71d0 ENA
sd c18t71d0s2-49 temp_6-01   c18t71d0s2 19290112
305152 161/0     c31t71d0 ENA
sd c21t71d0s2-49 temp_6-01   c21t71d0s2 19290112
305152 162/0     c34t71d0 ENA
sd c22t20d0s2-49 temp_6-01   c22t20d0s2 19290112
305152 163/0     c24t20d0 ENA
sd c24t20d0s2-49 temp_6-01   c24t20d0s2 19290112
305152 164/0     c33t20d0 ENA
sd c25t20d0s2-49 temp_6-01   c25t20d0s2 19290112
305152 165/0     c37t20d0 ENA
sd c26t20d0s2-49 temp_6-01   c26t20d0s2 19290112
305152 166/0     c36t20d0 ENA
sd c27t71d0s2-49 temp_6-01   c27t71d0s2 19290112
305152 167/0     c29t71d0 ENA
sd c28t20d0s2-49 temp_6-01   c28t20d0s2 19290112
305152 168/0     c39t20d0 ENA
sd c29t71d0s2-49 temp_6-01   c29t71d0s2 19290112
305152 169/0     c28t71d0 ENA
sd c30t71d0s2-49 temp_6-01   c30t71d0s2 19290112
305152 170/0     c27t71d0 ENA
sd c36t20d0s2-49 temp_6-01   c36t20d0s2 19290112
305152 171/0     c51t20d0 ENA
sd c38t71d0s2-49 temp_6-01   c38t71d0s2 19290112
305152 172/0     c50t71d0 ENA
sd c39t20d0s2-49 temp_6-01   c39t20d0s2 19290112
305152 173/0     c30t20d0 ENA
sd c43t71d0s2-49 temp_6-01   c43t71d0s2 19290112
305152 174/0     c48t71d0 ENA
sd c44t71d0s2-49 temp_6-01   c44t71d0s2 19290112
305152 175/0     c23t71d0 ENA
sd c45t20d0s2-49 temp_6-01   c45t20d0s2 19290112
305152 176/0     c22t20d0 ENA
sd c46t20d0s2-49 temp_6-01   c46t20d0s2 19290112
305152 177/0     c21t20d0 ENA
sd c10t104d0s2-49 temp_6-01  c10t104d0s2 19290112
305152 178/0    c15t104d0 ENA
sd c11t36d0s2-49 temp_6-01   c11t36d0s2 19290112
305152 179/0     c6t36d0  ENA
sd c12t68d0s2-49 temp_6-01   c12t68d0s2 19290112
305152 180/0     c10t68d0 ENA
sd c13t103d0s2-49 temp_6-01  c13t103d0s2 19290112
305152 181/0    c17t103d0 ENA
sd c14t68d0s2-49 temp_6-01   c14t68d0s2 19290112
305152 182/0     c19t68d0 ENA
sd c15t104d0s2-49 temp_6-01  c15t104d0s2 19290112
305152 183/0    c13t104d0 ENA
sd c16t36d0s2-49 temp_6-01   c16t36d0s2 19290112
305152 184/0     c7t36d0  ENA
sd c17t68d0s2-49 temp_6-01   c17t68d0s2 19290112
305152 185/0     c4t68d0  ENA
sd c18t104d0s2-49 temp_6-01  c18t104d0s2 19290112
305152 186/0    c31t104d0 ENA
sd c19t36d0s2-49 temp_6-01   c19t36d0s2 19290112
305152 187/0     c25t36d0 ENA
sd c20t68d0s2-49 temp_6-01   c20t68d0s2 19290112
305152 188/0     c32t68d0 ENA
sd c21t104d0s2-49 temp_6-01  c21t104d0s2 19290112
305152 189/0    c34t104d0 ENA
sd c22t103d0s2-49 temp_6-01  c22t103d0s2 19290112
305152 190/0    c24t103d0 ENA
sd c23t36d0s2-49 temp_6-01   c23t36d0s2 19290112
305152 191/0     c41t36d0 ENA
sd c24t103d0s2-49 temp_6-01  c24t103d0s2 19290112
305152 192/0    c33t103d0 ENA
sd c25t103d0s2-49 temp_6-01  c25t103d0s2 19290112
305152 193/0    c37t103d0 ENA
sd c26t103d0s2-49 temp_6-01  c26t103d0s2 19290112
305152 194/0    c36t103d0 ENA
sd c27t104d0s2-49 temp_6-01  c27t104d0s2 19290112
305152 195/0    c29t104d0 ENA
sd c28t103d0s2-49 temp_6-01  c28t103d0s2 19290112
305152 196/0    c39t103d0 ENA
sd c29t104d0s2-49 temp_6-01  c29t104d0s2 19290112
305152 197/0    c28t104d0 ENA
sd c30t104d0s2-49 temp_6-01  c30t104d0s2 19290112
305152 198/0    c27t104d0 ENA
sd c31t68d0s2-49 temp_6-01   c31t68d0s2 19290112
305152 199/0     c38t68d0 ENA
sd c32t36d0s2-49 temp_6-01   c32t36d0s2 19290112
305152 200/0     c40t36d0 ENA
sd c33t68d0s2-49 temp_6-01   c33t68d0s2 19290112
305152 201/0     c42t68d0 ENA
sd c34t36d0s2-49 temp_6-01   c34t36d0s2 19290112
305152 202/0     c46t36d0 ENA
sd c35t68d0s2-49 temp_6-01   c35t68d0s2 19290112
305152 203/0     c43t68d0 ENA
sd c36t103d0s2-49 temp_6-01  c36t103d0s2 19290112
305152 204/0    c51t103d0 ENA
sd c37t68d0s2-49 temp_6-01   c37t68d0s2 19290112
305152 205/0     c44t68d0 ENA
sd c38t104d0s2-49 temp_6-01  c38t104d0s2 19290112
305152 206/0    c50t104d0 ENA
sd c39t103d0s2-49 temp_6-01  c39t103d0s2 19290112
305152 207/0    c30t103d0 ENA
sd c40t36d0s2-49 temp_6-01   c40t36d0s2 19290112
305152 208/0     c26t36d0 ENA
sd c41t36d0s2-49 temp_6-01   c41t36d0s2 19290112
305152 209/0     c47t36d0 ENA
sd c42t68d0s2-49 temp_6-01   c42t68d0s2 19290112
305152 210/0     c49t68d0 ENA
sd c43t104d0s2-49 temp_6-01  c43t104d0s2 19290112
305152 211/0    c48t104d0 ENA
sd c44t104d0s2-49 temp_6-01  c44t104d0s2 19290112
305152 212/0    c23t104d0 ENA
sd c45t103d0s2-49 temp_6-01  c45t103d0s2 19290112
305152 213/0    c22t103d0 ENA
sd c46t103d0s2-49 temp_6-01  c46t103d0s2 19290112
305152 214/0    c21t103d0 ENA
sd c47t36d0s2-49 temp_6-01   c47t36d0s2 19290112
305152 215/0     c3t36d0  ENA
sd c48t36d0s2-49 temp_6-01   c48t36d0s2 19290112
305152 216/0     c20t36d0 ENA
sd c49t68d0s2-49 temp_6-01   c49t68d0s2 19290112
305152 217/0     c8t68d0  ENA
sd c10t72d0s2-49 temp_6-01   c10t72d0s2 19290112
305152 218/0     c15t72d0 ENA
sd c13t21d0s2-49 temp_6-01   c13t21d0s2 19290112
305152 219/0     c17t21d0 ENA
```

Appendix C. Query Text and Query Output

```
==================================================
```

qual01.v1

```
==================================================
Begin Execution at Sat Apr  7 08:05:06 2001


-- using default substitutions

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

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


4 rows processed.
Statement Processed in 13.35 seconds.

Ended Executing this Query at Sat Apr  7 08:05:19 2001


Query Started at 986655906.32
Query Ended at 986655919.68
Query Processed in 13.35 seconds


SQL statements processed: 1
Queries processed: 1

==================================================
```

qual02.v1

```
==================================================
Begin Execution at Sat Apr  7 08:05:19 2001


-- using default substitutions

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

S_ACCTBAL              S_NAME
N_NAME
P_PARTKEY              P_MFGR
S_ADDRESS                                 S_PHONE
S_COMMENT
9938.53                Supplier#000005359
UNITED KINGDOM
185358.00              Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04                  33-429-790-
6131
blithely silent pinto beans are furiously. slyly final
deposits acros
9937.84                Supplier#000005969
ROMANIA
108438.00              Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa         29-520-692-
3537
carefully slow deposits use furiously. slyly ironic
platelets above the ironic
9936.22                Supplier#000005250
UNITED KINGDOM
249.00                 Manufacturer#4
B3rqp0xbSEim4Mpy2RH J                     33-320-228-
2957
blithely special packages are. stealthily express
deposits across the closely final instructi
9923.77                Supplier#000002324
GERMANY
29821.00               Manufacturer#4
y3OD9UywSTOk                              17-779-299-
1839
quickly express packages breach quiet pinto beans.
requ
9871.22                Supplier#000006373
GERMANY
43868.00               Manufacturer#5
J8fcXWsTqM                                17-813-485-
8637
never silent deposits integrate furiously blit
9870.78                Supplier#000001286
GERMANY
81285.00               Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH     17-516-924-
```

4574
final theodolites cajole slyly special,
9870.78 Supplier#000001286
GERMANY
181285.00 Manufacturer#4
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH 17-516-924-
4574
final theodolites cajole slyly special,
9852.52 Supplier#000008973
RUSSIA
18972.00 Manufacturer#2
t5L67YdBYYH6o,Vz24jpDyQ9 32-188-594-
7038
quickly regular instructions wake-- carefully unusual
braids into the expres
9847.83 Supplier#000008097
RUSSIA
130557.00 Manufacturer#2
xMe97bpE69NzdwLoX 32-375-640-
3593
slyly regular dependencies sleep slyly furiously
express dep
9847.57 Supplier#000006345
FRANCE
86344.00 Manufacturer#1
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-
7945
silent pinto beans should have to snooze carefully
along the final reques
9847.57 Supplier#000006345
FRANCE
173827.00 Manufacturer#2
VSt3rzk3qG698u6ld8HhOByvrTcSTSvQlDQDag 16-886-766-
7945
silent pinto beans should have to snooze carefully
along the final reques
9836.93 Supplier#000007342
RUSSIA
4841.00 Manufacturer#4
JOlK7C1,7xrEZSSOw 32-399-414-
5385
final accounts haggle. bold accounts are furiously
dugouts. furiously silent asymptotes are slyly
9817.10 Supplier#000002352
RUSSIA
124815.00 Manufacturer#2
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-
1437
blithely pending packages across the ironic accounts
grow slyly after the furiously
9817.10 Supplier#000002352
RUSSIA
152351.00 Manufacturer#3
4LfoHUZjgjEbAKw TgdKcgOc4D4uCYw 32-551-831-
1437
blithely pending packages across the ironic accounts
grow slyly after the furiously
9739.86 Supplier#000003384
FRANCE
138357.00 Manufacturer#2
o,Z3v4POifevE k9U1b 6J1ucX,I 16-494-913-
5925
slyly ironic theodolites hag
9721.95 Supplier#000008757
UNITED KINGDOM
156241.00 Manufacturer#3
Atg6GnM4dT2 33-821-407-
2995
ironic, even dolphins above the furiously ironic foxes
sleep slyly around the caref
9681.33 Supplier#000008406
RUSSIA
78405.00 Manufacturer#1
,qUuXcftUl 32-139-873-
8571
furiously even deposits affix thinly special
theodolites. furiou
9643.55 Supplier#000005148
ROMANIA
107617.00 Manufacturer#1
kT4ciVFslx9z4s79p Js825 29-252-617-
4850
doggedly even ideas boost furiously against the
furiously express
9624.82 Supplier#000001816
FRANCE
34306.00 Manufacturer#3
e7vab91vLJPWxxZnewmnDBpDmxYHrb 16-392-237-
6726
blithely regular accounts cajole furiously. regular
9624.78 Supplier#000009658
ROMANIA
189657.00 Manufacturer#1
oE9uBgEfSS4opIcepXyAYM,x 29-748-876-
2014
regular deposits haggle. furiously express asympto
9612.94 Supplier#000003228
ROMANIA
120715.00 Manufacturer#2
KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB 29-325-784-
8187
carefully pending accounts serve. furiously close
deposits boost slyly. q
9612.94 Supplier#000003228
ROMANIA
198189.00 Manufacturer#4
KDdpNKN3cWu7ZSrbdqp7AfSLxx,qWB 29-325-784-
8187
carefully pending accounts serve. furiously close
deposits boost slyly. q
9571.83 Supplier#000004305
ROMANIA
179270.00 Manufacturer#2
qNHZ7WmCzygwMPRDO9Ps 29-973-481-
1831
furiously final deposits
9558.10 Supplier#000003532
UNITED KINGDOM
88515.00 Manufacturer#4
EOeuiiOn21OVpTlGguufFDFsbN1p0lhpxHp 33-152-301-
2164
daring, sly accounts breach about th
9492.79 Supplier#000005975
GERMANY
25974.00 Manufacturer#5
S6mIiCTx82z7lV 17-992-579-
4839
always pending packages boost slyly.
9461.05 Supplier#000002536
UNITED KINGDOM
20033.00 Manufacturer#1
8mmGbyzaU 7ZS2wJumTibypncu9pNkDc4FYA 33-556-973-
5522
even foxes are quickly furiously express requests.
packages
9453.01 Supplier#000000802
ROMANIA
175767.00 Manufacturer#1
,6HYXb4uaHITmtMBj4Ak57Pd 29-342-882-
6463
final, regular packages across the slowly regular
packag
9408.65 Supplier#000007772
UNITED KINGDOM
117771.00 Manufacturer#4
AiC5YAH,gdu0i7 33-152-491-
1126
blithely final ideas sleep carefully. requests are
9359.61 Supplier#000004856
ROMANIA
62349.00 Manufacturer#5
HYogcF3Jb yh1 29-334-870-
9731
carefully unusual packages sleep carefully even ideas.
dogged accoun
9357.45 Supplier#000006188
UNITED KINGDOM
138648.00 Manufacturer#1
g801,ssP8wpTk4Hm 33-583-607-
1633
carefully regular deposits wake carefully furiously
even i
9352.04 Supplier#000003439
GERMANY
170921.00 Manufacturer#4
qYPDgoiBGhCYxjgC 17-128-996-
4650
fluffily regular pinto beans wake. unusual, final
ideas c
9312.97 Supplier#000007807
RUSSIA
90279.00 Manufacturer#5
oGYMPCk9XHGB2PBfKRnHA 32-673-872-
5854
unusual asymptotes above the
9312.97 Supplier#000007807
RUSSIA
100276.00 Manufacturer#5
oGYMPCk9XHGB2PBfKRnHA 32-673-872-
5854

unusual asymptotes above the
9280.27 Supplier#000007194
ROMANIA
47193.00 Manufacturer#3
zhRUQkBSrFYxIAXTfInj vyGRQjeK 29-318-454-2133
slyly ironic requests despite the unusual ins
9274.80 Supplier#000008854
RUSSIA
76346.00 Manufacturer#3
1xhLoOUM7I3mZ1mKnerw OSqdbb4QbGa 32-524-148-5221
ruthlessly ironic instructions along the regular, furious requests integrate car
9249.35 Supplier#000003973
FRANCE
26466.00 Manufacturer#1
d18GiDsL6Wm2IsGXM,RZf1jCsgZAOjNYVThTRP4 16-722-866-1658
quickly ironic sauternes use b
9249.35 Supplier#000003973
FRANCE
33972.00 Manufacturer#1
d18GiDsL6Wm2IsGXM,RZf1jCsgZAOjNYVThTRP4 16-722-866-1658
quickly ironic sauternes use b
9208.70 Supplier#000007769
ROMANIA
40256.00 Manufacturer#5
rsimdze 5o9P Ht7xS 29-964-424-9649
furiously ruthless epitaphs among the furiously regular accounts use slowly fluffily ev
9201.47 Supplier#000009690
UNITED KINGDOM
67183.00 Manufacturer#5
CB BnUTlmi5zdeEl7R7 33-121-267-9529
blithely unusual accounts integrate slyly. platelets
9192.10 Supplier#000000115
UNITED KINGDOM
85098.00 Manufacturer#3
nJ 2t0f7Ve,wL1,6WzGBJLNBUCKlsV 33-597-248-1220
slyly bold pinto beans boost across the furiously regular packages. carefully regu
9189.98 Supplier#000001226
GERMANY
21225.00 Manufacturer#4
qsLCqSvLyZfuXIpjz 17-725-903-1381
final, express instruction
9128.97 Supplier#000004311
RUSSIA
146768.00 Manufacturer#5
I8IjnXd7NSJRs594RxsRR0 32-155-440-7120
regular pinto beans sleep ca
9104.83 Supplier#000008520
GERMANY
150974.00 Manufacturer#4
RqRVDgD0ER J9 b41vR2,3 17-728-804-1793
deposits sleep carefully e
9101.00 Supplier#000005791
ROMANIA
128254.00 Manufacturer#5
zub2zCV,jhHPPQqi,P2INAjE1zI n66cOEoXFG 29-549-251-5384
carefully ironic packages after the
9094.57 Supplier#000004582
RUSSIA
39575.00 Manufacturer#1
WB0XkCSG3r,mnQ n,h9VIxjjr9ARHFvKgMDf 32-587-577-1351
asymptotes above the slyly even requests haggle furiously about the regular accounts
8996.87 Supplier#000004702
FRANCE
102191.00 Manufacturer#5
8XVcQK23akp 16-811-269-8946
stealthy requests haggle c
8996.14 Supplier#000009814
ROMANIA
139813.00 Manufacturer#2
af0O5pg83lPU4IDVmEylXZVqYZQzSDlYLAmR 29-995-571-8781
ironic theodolites are evenly unusual requests-- pending pinto beans across the in
8968.42 Supplier#000010000
ROMANIA
119999.00 Manufacturer#5
aTGLEusCiL4F PDBdv665XBJhPyCOB0i 29-578-432-2146
furiously final ideas believe furiously. furiously final ideas
8936.82 Supplier#000007043
UNITED KINGDOM
109512.00 Manufacturer#1
FVajceZInZdbJE6Z9XsRUxrUEpiwHDrOXi,1Rz 33-784-177-8208
furiously regular excuses wake after the blithely special pinto beans? even instructions sl
8929.42 Supplier#000008770
FRANCE
173735.00 Manufacturer#4
R7cG26TtXrHAP9 HckhfRi 16-242-746-9248
final accounts sleep furiously. blithely ironic foxes wake boldly across the furiously s
8920.59 Supplier#000003967
ROMANIA
26460.00 Manufacturer#1
eHoAXe62SY9 29-194-731-3944
quickly even requests should have to affix blithely--fur
8920.59 Supplier#000003967
ROMANIA
173966.00 Manufacturer#2
eHoAXe62SY9 29-194-731-3944
quickly even requests should have to affix blithely--fur
8913.96 Supplier#000004603
UNITED KINGDOM
137063.00 Manufacturer#2
OUzlvMUr7n,utLxmPNeYKSf3T24OXskxB5 33-789-255-7342
slyly ironic packages detect furious accounts. ironic de
8877.82 Supplier#000007967
FRANCE
167966.00 Manufacturer#5
A3pi1BARM4nx6R,qrwFoRPU 16-442-147-9345
final deposits after the silent deposits ha
8862.24 Supplier#000003323
ROMANIA
73322.00 Manufacturer#3
W9 lYcsC9FwBqk3ItL 29-736-951-3710
unusual, pending theodolites integrate furiously slyly even pinto beans. unusual sheaves sleep befor
8841.59 Supplier#000005750
ROMANIA
100729.00 Manufacturer#5
Erx3lAgu0g62iaHF9x50uMH4EgeN9hEG 29-344-502-5481
excuses after the blithely regular packages mold carefully deposits. regular a
8781.71 Supplier#000003121
ROMANIA
13120.00 Manufacturer#5
wNqTogx238ZYCamFb,50v,bj 4IbNFW9Bvw1xP 29-707-291-5144
packages are quickly after the final, even packages. furiously regular
8754.24 Supplier#000009407
UNITED KINGDOM
179406.00 Manufacturer#4
CHRCbkaWcf5B 33-903-970-9604
regular dependencies haggle across the carefully bold
8691.06 Supplier#000004429
UNITED KINGDOM
126892.00 Manufacturer#2
k,BQms5UhoAF1B2Asi,fLib 33-964-337-5038
quickly special foxes against the furiously silent platelets wake quickly after t
8655.99 Supplier#000006330
RUSSIA
193810.00 Manufacturer#2
UozlaENr0ytKe2w6CeIEWFWn iO3S8Rae7Ou 32-561-198-3705
blithely even packages alongside
8638.36 Supplier#000002920
RUSSIA

75398.00 Manufacturer#1
Je2a8bszf3L 32-122-621-7549
express deposits wake. furiously silent requests wake carefully silent instru
8638.36 Supplier#000002920
RUSSIA
170402.00 Manufacturer#3
Je2a8bszf3L 32-122-621-7549
express deposits wake. furiously silent requests wake carefully silent instru
8607.69 Supplier#000006003
UNITED KINGDOM
76002.00 Manufacturer#2
EH9wADcEiuenM0NR08zDwMidw,52Y2RyILEiA 33-416-807-5206
always special foxes wake slyly bold, ironic accounts. ironic instructions affix carefull
8569.52 Supplier#000005936
RUSSIA
5935.00 Manufacturer#5
jXaNZ6vwnEWJ2ksLZJpjtgt0bY2a3AU 32-644-251-7916
packages sleep furiously. special requests about the fluffily even accounts detect
8564.12 Supplier#000000033
GERMANY
110032.00 Manufacturer#1
gfeKpYw3400L0SDywXA6Ya1Qmq1w6YB9f3R 17-138-897-9374
ironic instructions are. special pearls above
8553.82 Supplier#000003979
ROMANIA
143978.00 Manufacturer#4
BfmVhCAnCMY3jzpjUMy4CNWs9 HzpdQR7INJU 29-124-646-4897
express, ironic pinto beans cajole around the express, even packages. qu
8517.23 Supplier#000009529
RUSSIA
37025.00 Manufacturer#5
e44R8o7JAIS9iMcr 32-565-297-8775
furiously silent requests cajole furiously furiously ironic foxes. slyly express p
8517.23 Supplier#000009529
RUSSIA
59528.00 Manufacturer#2
e44R8o7JAIS9iMcr 32-565-297-8775
furiously silent requests cajole furiously furiously ironic foxes. slyly express p
8503.70 Supplier#000006830
RUSSIA
44325.00 Manufacturer#4
BC4WFCYRUZyaIgchU 4S 32-147-878-5069
quickly regular excuses detect evenly around
8457.09 Supplier#000009456
UNITED KINGDOM
19455.00 Manufacturer#1
7SBhZs8gP1cJjT0Qf433YBk 33-858-440-4349
carefully final accounts sleep blithely special foxes. slyly regular pinto beans alon
8441.40 Supplier#000003817
FRANCE
141302.00 Manufacturer#2
hU3fz3xL78 16-339-356-5115
blithely blithe ideas are
8432.89 Supplier#000003990
RUSSIA
191470.00 Manufacturer#1
wehBBp1RQbfxAYDASS75MsywmsKHRVdkrvNe6m 32-839-509-9301
final requests along the blithely ironic packages kindle against the carefully fina
8431.40 Supplier#000002675
ROMANIA
5174.00 Manufacturer#1
HJFStOu9R5NGPOegKhgbzBdyvrG2yh8w 29-474-643-1443
express, final deposits cajole carefully. stealthily unusual requests
8407.04 Supplier#000005406
RUSSIA
162889.00 Manufacturer#4
j7 gYF5RW8DC5UrjKC 32-626-152-4621
quickly final sheaves boost. car
8386.08 Supplier#000008518
FRANCE
36014.00 Manufacturer#3
2jqzqqAVe9crMVGP,n9nTsQXulNLTUYoJjEDcqWV 16-618-780-7481
slyly ironic theodolites are slyly. dogged, pendin
8376.52 Supplier#000005306
UNITED KINGDOM
190267.00 Manufacturer#5
9t8Y8 QqSIsoADPt6NLdk,TP5zyRx41oBUlgoGc9 33-632-514-7931
furiously even instructions integrate during the furiously regular re
8348.74 Supplier#000008851
FRANCE
66344.00 Manufacturer#4
nWxi7GwEbjhw1 16-796-240-2472
ironic instructions nag slyly against the slyly even theodolites. requests alongside of
8338.58 Supplier#000007269
FRANCE
17268.00 Manufacturer#4
ZwhJSwABUoiB04,3 16-267-277-4365
ruthlessly regular asymptotes a
8328.46 Supplier#000001744
ROMANIA
69237.00 Manufacturer#5
oLo3fV64q2,FKHa3p,qHnS7Yzv,ps8 29-330-728-5873
blithely silent excuses are slyly above the furiously even courts
8307.93 Supplier#000003142
GERMANY
18139.00 Manufacturer#1
dqblvV8dCNAorGlJ 17-595-447-6026
theodolites sleep blithely carefully regular warhorses. slyly regular ins
8231.61 Supplier#000009558
RUSSIA
192000.00 Manufacturer#2
mcdgen,yT1iJDHDS5fV 32-762-137-5858
slyly regular theodolites sleep fluffily express depos
8152.61 Supplier#000002731
ROMANIA
15227.00 Manufacturer#4
 nluXJCuY1tu 29-805-463-2030
gifts use. slyly silent ideas are carefully beneath the silent instructions. slyly sil
8109.09 Supplier#000009186
FRANCE
99185.00 Manufacturer#1
wgfosrVPexl9pEXWywaqlBMDYYf 16-668-570-1402
quickly pending requests are blithely along the ironic, final requests; instr
8102.62 Supplier#000003347
UNITED KINGDOM
18344.00 Manufacturer#5
m CtXS2S16i 33-454-274-8532
packages grow special orbits. regular theodolites about the carefully pe
8046.07 Supplier#000008780
FRANCE
191222.00 Manufacturer#3
AczzuE0UK9osj ,Lx0Jmh 16-473-215-6395
regular epitaphs integrate slyly.
8042.09 Supplier#000003245
RUSSIA
135705.00 Manufacturer#4
Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872
carefully regular instructions integrate blithely silent foxes. furiously express instructions haggl
8042.09 Supplier#000003245
RUSSIA
150729.00 Manufacturer#1
Dh8Ikg39onrbOL4DyTfGw8a9oKUX3d9Y 32-836-132-8872
carefully regular instructions integrate blithely silent foxes. furiously express instructions haggl
7992.40 Supplier#000006108

```
FRANCE
118574.00               Manufacturer#1
8tBydnTDwUqfBfFV4l3                          16-974-998-
8937
regular pinto beans are after
7980.65                 Supplier#000001288
FRANCE
13784.00                Manufacturer#4
zE,7HgVPrCn                                  16-646-464-
8247
unusual pinto beans cajole furiously according t
7950.37                 Supplier#000008101
GERMANY
33094.00                Manufacturer#5
kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj            17-627-663-
8014
quickly regular requests are furiously. pending
deposits wake
7937.93                 Supplier#000009012
ROMANIA
83995.00                Manufacturer#2
iUiTziH,Ek3i4lwSgunXMgrcTzwdb                29-250-925-
9690
blithely bold ideas haggle quickly final, regular
request
7914.45                 Supplier#000001013
RUSSIA
125988.00               Manufacturer#2
riRcntps4KEDtYScjpMIWeYF6mNnR                32-194-698-
3365
final, ironic theodolites alongside of the ironic
7912.91                 Supplier#000004211
GERMANY
159180.00               Manufacturer#5
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG            17-266-947-
7315
final requests integrate slyly above the silent, even
7912.91                 Supplier#000004211
GERMANY
184210.00               Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG            17-266-947-
7315
final requests integrate slyly above the silent, even
7894.56                 Supplier#000007981
GERMANY
85472.00                Manufacturer#4
NSJ96vMROAbeXP                               17-963-404-
3760
regular, even theodolites integrate carefully. bold,
special theodolites are slyly fluffily iron
7887.08                 Supplier#000009792
GERMANY
164759.00               Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz          17-988-938-
4296
pending, ironic packages sleep among the carefully
ironic accounts. quickly final accounts
7871.50                 Supplier#000007206
RUSSIA
104695.00               Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW                      32-432-452-
7731
furiously dogged pinto beans cajole. bold, express
notornis until the slyly pending
7852.45                 Supplier#000005864
RUSSIA
8363.00                 Manufacturer#4
WCNfBPZeSXh3h,c                              32-454-883-
3821
blithely regular deposits
7850.66                 Supplier#000001518
UNITED KINGDOM
86501.00                Manufacturer#1
ONda3YJiHKJOC                                33-730-383-
3892
furiously final accounts wake carefully idle requests.
even dolphins wake acc
7843.52                 Supplier#000006683
FRANCE
11680.00                Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy                16-464-517-
8943
carefully bold accounts doub


100 rows processed.
Statement Processed in 7.66 seconds.

Ended Executing this Query at Sat Apr  7 08:05:27 2001

Query Started at 986655919.94
Query Ended at 986655927.60
Query Processed in 7.66 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual03.v1

```
====================================================
Begin Execution at Sat Apr  7 08:05:27 2001


-- using default substitutions

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

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


10 rows processed.
Statement Processed in 7.85 seconds.

Ended Executing this Query at Sat Apr  7 08:05:35 2001


Query Started at 986655927.88
Query Ended at 986655935.73
Query Processed in 7.85 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual04.v1

```
====================================================
Begin Execution at Sat Apr  7 08:05:35 2001


-- using default substitutions
```

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

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


5 rows processed.
Statement Processed in 7.78 seconds.

Ended Executing this Query at Sat Apr  7 08:05:43 2001


Query Started at 986655936.00
Query Ended at 986655943.78
Query Processed in 7.78 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual05.v1

```
====================================================
Begin Execution at Sat Apr  7 08:05:44 2001


-- using default substitutions

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

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


5 rows processed.
Statement Processed in 10.64 seconds.

Ended Executing this Query at Sat Apr  7 08:05:54 2001


Query Started at 986655944.04
Query Ended at 986655954.69
Query Processed in 10.64 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual06.v1

```
====================================================
Begin Execution at Sat Apr  7 08:05:54 2001


-- using default substitutions

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

REVENUE
123141078.23


1 row

====================================================
```

qual07.v1

```
====================================================
Begin Execution at Sat Apr  7 08:05:56 2001


-- using default substitutions

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

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


4 rows processed.
Statement Processed in 8.63 seconds.

Ended Executing this Query at Sat Apr  7 08:06:05 2001


Query Started at 986655956.59
Query Ended at 986655965.22
Query Processed in 8.63 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual08.v1

```
====================================================
Begin Execution at Sat Apr  7 08:06:05 2001


-- using default substitutions

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

O_YEAR                 MKT_SHARE
1995.00                0.03
1996.00                0.04


2 rows processed.
Statement Processed in 14.06 seconds.

Ended Executing this Query at Sat Apr  7 08:06:19 2001


Query Started at 986655965.49
Query Ended at 986655979.55
Query Processed in 14.06 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual09.v1

```
====================================================
Begin Execution at Sat Apr  7 08:06:19 2001


-- using default substitutions

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

NATION                     O_YEAR
SUM_PROFIT
ALGERIA                    1998.00
31342867.23
ALGERIA                    1997.00
57138193.02
ALGERIA                    1996.00
56140140.13
ALGERIA                    1995.00
53051469.65
ALGERIA                    1994.00
53867582.13
ALGERIA                    1993.00
54942718.13
ALGERIA                    1992.00
54628034.71
ARGENTINA                  1998.00
30211185.71
ARGENTINA                  1997.00
50805741.75
ARGENTINA                  1996.00
51923746.58
ARGENTINA                  1995.00
49298625.77
ARGENTINA                  1994.00
50835610.11
ARGENTINA                  1993.00
51646079.18
ARGENTINA                  1992.00
50410314.99
BRAZIL                     1998.00
27217924.38
BRAZIL                     1997.00
48378669.20
BRAZIL                     1996.00
50482870.36
BRAZIL                     1995.00
47623383.63
BRAZIL                     1994.00
47840165.73
```

BRAZIL 1993.00 49054694.04
BRAZIL 1992.00 48667639.08
CANADA 1998.00 30379833.77
CANADA 1997.00 50465052.31
CANADA 1996.00 52560501.39
CANADA 1995.00 52375332.81
CANADA 1994.00 52600364.66
CANADA 1993.00 52644504.07
CANADA 1992.00 53932871.70
CHINA 1998.00 31075466.16
CHINA 1997.00 50551874.45
CHINA 1996.00 51039293.88
CHINA 1995.00 49287534.62
CHINA 1994.00 50851090.07
CHINA 1993.00 54229629.83
CHINA 1992.00 52400529.37
EGYPT 1998.00 29054433.39
EGYPT 1997.00 50627611.45
EGYPT 1996.00 49542212.84
EGYPT 1995.00 48311550.32
EGYPT 1994.00 49790644.74
EGYPT 1993.00 48904292.97
EGYPT 1992.00 49434932.62
ETHIOPIA 1998.00 28040717.27
ETHIOPIA 1997.00 47455009.87
ETHIOPIA 1996.00 46491097.57
ETHIOPIA 1995.00 46804449.30
ETHIOPIA 1994.00 48516143.92
ETHIOPIA 1993.00 46551891.56
ETHIOPIA 1992.00 44934648.64
FRANCE 1998.00 32226407.84
FRANCE 1997.00 47121485.86
FRANCE 1996.00 47263135.50
FRANCE 1995.00 47275997.57
FRANCE 1994.00 47067209.33
FRANCE 1993.00 51163370.11
FRANCE 1992.00 47846235.33
GERMANY 1998.00 28624942.66
GERMANY 1997.00 49309074.88
GERMANY 1996.00 49918683.17
GERMANY 1995.00 52650718.72
GERMANY 1994.00 50346900.42
GERMANY 1993.00 50991895.81
GERMANY 1992.00 48274126.10
INDIA 1998.00 29943144.35
INDIA 1997.00 50665453.23
INDIA 1996.00 50283092.29
INDIA 1995.00 50006774.64
INDIA 1994.00 48995190.76
INDIA 1993.00 50286902.85
INDIA 1992.00 50850329.40
INDONESIA 1998.00 27672340.00
INDONESIA 1997.00 50512145.73
INDONESIA 1996.00 51653060.12
INDONESIA 1995.00 51508779.59
INDONESIA 1994.00 52817950.32
INDONESIA 1993.00 47959994.96
INDONESIA 1992.00 51776605.03
IRAN 1998.00 29065736.24
IRAN 1997.00 50042063.05
IRAN 1996.00 50926653.19
IRAN 1995.00 51249667.65
IRAN 1994.00 50337085.87
IRAN 1993.00 51730763.49
IRAN 1992.00 49955856.56
IRAQ 1998.00 31624551.00
IRAQ 1997.00 55121749.02
IRAQ 1996.00 55897663.79
IRAQ 1995.00 54815472.52
IRAQ 1994.00 54408516.13
IRAQ 1993.00 53633167.98
IRAQ 1992.00 55891939.34
JAPAN 1998.00 27934179.67
JAPAN 1997.00 44517162.55
JAPAN 1996.00 42545606.12
JAPAN 1995.00 43749356.40
JAPAN 1994.00 44840243.07
JAPAN 1993.00 44660015.53
JAPAN 1992.00 45410249.12
JORDAN 1998.00 26901488.58
JORDAN 1997.00 45471878.41
JORDAN 1996.00 46794325.79
JORDAN 1995.00 45178828.58
JORDAN 1994.00 45333636.51
JORDAN 1993.00 47971496.10
JORDAN 1992.00 44717239.18
KENYA 1998.00 28597614.34
KENYA 1997.00 47949733.73
KENYA 1996.00 46886924.62
KENYA 1995.00 46072338.76

```
KENYA                     1994.00
45772061.17
KENYA                     1993.00
46308728.23
KENYA                     1992.00
47257780.84
MOROCCO                   1998.00
26732115.58
MOROCCO                   1997.00
45637304.25
MOROCCO                   1996.00
45558221.75
MOROCCO                   1995.00
47851318.89
MOROCCO                   1994.00
46272172.94
MOROCCO                   1993.00
46764326.18
MOROCCO                   1992.00
48122783.58
MOZAMBIQUE                1998.00
30712392.01
MOZAMBIQUE                1997.00
50316528.76
MOZAMBIQUE                1996.00
51640320.25
MOZAMBIQUE                1995.00
50693774.51
MOZAMBIQUE                1994.00
49253277.63
MOZAMBIQUE                1993.00
49153016.54
MOZAMBIQUE                1992.00
48247551.85
PERU                      1998.00
29326102.32
PERU                      1997.00
49753780.40
PERU                      1996.00
50935170.29
PERU                      1995.00
53309883.41
PERU                      1994.00
50643531.80
PERU                      1993.00
51584622.00
PERU                      1992.00
47523899.05
ROMANIA                   1998.00
30368667.40
ROMANIA                   1997.00
50365683.85
ROMANIA                   1996.00
49598999.01
ROMANIA                   1995.00
47537642.87
ROMANIA                   1994.00
51455283.01
ROMANIA                   1993.00
50407136.89
ROMANIA                   1992.00
48185385.13
RUSSIA                    1998.00
28322384.03
RUSSIA                    1997.00
50106685.18
RUSSIA                    1996.00
51753342.43
RUSSIA                    1995.00
49215820.36
RUSSIA                    1994.00
52205666.44
RUSSIA                    1993.00
51860230.03
RUSSIA                    1992.00
53251677.15
SAUDI ARABIA              1998.00
31541259.81
SAUDI ARABIA              1997.00
52438750.81
SAUDI ARABIA              1996.00
52543737.82
SAUDI ARABIA              1995.00
52938696.53
SAUDI ARABIA              1994.00
51389601.97
SAUDI ARABIA              1993.00
52937508.88
SAUDI ARABIA              1992.00
54843459.64
UNITED KINGDOM            1998.00
28494874.00
UNITED KINGDOM            1997.00
49381810.90
UNITED KINGDOM            1996.00
51386853.96
UNITED KINGDOM            1995.00
51509586.79
UNITED KINGDOM            1994.00
48086499.71
UNITED KINGDOM            1993.00
49166827.22
UNITED KINGDOM            1992.00
49349122.08
UNITED STATES             1998.00
25126238.95
UNITED STATES             1997.00
50077306.42
UNITED STATES             1996.00
48048649.47
UNITED STATES             1995.00
48809032.42
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

175 rows processed.
Statement Processed in 15.38 seconds.

Ended Executing this Query at Sat Apr  7 08:06:35 2001

Query Started at 986655979.83
Query Ended at 986655995.21
Query Processed in 15.38 seconds


SQL statements processed: 1
Queries processed: 1

==================================================
```

qual10.v1

```
==================================================
Begin Execution at Sat Apr  7 08:06:35 2001


-- using default substitutions

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

```
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20

C_CUSTKEY              C_NAME
REVENUE
C_ACCTBAL              N_NAME
C_ADDRESS                                C_PHONE
C_COMMENT
57040.00               Customer#000057040
734235.25
632.87                 JAPAN
Eioyzjf4pp                               22-895-641-
3466
requests sleep blithely about the furiously i
143347.00              Customer#000143347
721002.69
2557.47                EGYPT
1aReFYv,Kw4                              14-742-935-
3718
fluffily bold excuses haggle finally after the u
60838.00               Customer#000060838
679127.31
2454.77                BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE             12-913-494-
9813
furiously even pinto beans integrate under the
ruthless foxes; ironic, even dolphins across the slyl
101998.00              Customer#000101998
637029.57
3790.89                UNITED KINGDOM
01c9CILnNtfOQYmZj                        33-593-865-
6378
accounts doze blithely! enticing, final deposits sleep
blithely special accounts. slyly express accounts pla
125341.00              Customer#000125341
633508.09
4983.51                GERMANY
S29ODD6bceU8QSuuEJznkNaK                 17-582-695-
5962
quickly express requests wake quickly blithely
25501.00               Customer#000025501
620269.78
7725.04                ETHIOPIA
  W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ     15-874-808-
6793
quickly special requests sleep evenly among the
special deposits. special deposi
115831.00              Customer#000115831
596423.87
5098.10                FRANCE
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-
3788
carefully bold excuses sleep alongside of the thinly
idle
84223.00               Customer#000084223
594998.02
528.65                 UNITED KINGDOM
nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA       33-442-824-
8191
pending, final ideas haggle final requests. unusual,
regular asymptotes affix according to the even foxes.
54289.00               Customer#000054289
585603.39
5583.02                IRAN
vXCxoCsU0Bad5JQI ,oobkZ                  20-834-292-
4707
express requests sublate blithely regular requests.
regular, even ideas solve.
39922.00               Customer#000039922
584878.11
7321.11                GERMANY
Zgy4s50l2GKN4pLDPBU8m342gIw6R            17-147-757-
8036
even pinto beans haggle. slyly bold accounts inte
6226.00                Customer#000006226
576783.76
2230.09                UNITED KINGDOM
8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g,        33-657-701-
3391
quickly final requests against the regular
instructions wake blithely final instructions. pa
922.00                 Customer#000000922
576767.53
3869.25                GERMANY
Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq          17-945-916-
9648
boldly final requests cajole blith
147946.00              Customer#000147946
576455.13
2030.13                ALGERIA
iANyZHjqhyy7Ajah0pTrYyhJ                 10-886-956-
3143
furiously even accounts are blithely above the
furiousl
115640.00              Customer#000115640
569341.19
6436.10                ARGENTINA
Vtgfia9qI 7EpHgecU1X                     11-411-543-
4901
final instructions are slyly according to the
73606.00               Customer#000073606
568656.86
1785.67                JAPAN
xuR0Tro5yChDfOCrjkd2ol                   22-437-653-
6966
furiously bold orbits about the furiously busy
requests wake across the furiously quiet theodolites.
d
110246.00              Customer#000110246
566842.98
7763.35                VIETNAM
7KzflgX MDOq7sOkI                        31-943-426-
9837
dolphins sleep blithely among the slyly final
142549.00              Customer#000142549
563537.24
5085.99                INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9           19-955-562-
2398
regular, unusual dependencies boost slyly; ironic
attainments nag fluffily into the unusual packages?
146149.00              Customer#000146149
557254.99
1791.55                ROMANIA
s87fvzFQpU                               29-744-164-
6487
silent, unusual requests detect quickly slyly regul
52528.00               Customer#000052528
556397.35
551.79                 ARGENTINA
NFztyTOR10UOJ                            11-208-192-
3205
unusual requests detect. slyly dogged theodolites use
slyly. deposit
23431.00               Customer#000023431
554269.54
3381.86                ROMANIA
HgiV0phqhaIa9aydNoIlb                    29-915-458-
2654
instructions nag quickly. furiously bold accounts
cajol


20 rows processed.
Statement Processed in 9.73 seconds.

Ended Executing this Query at Sat Apr  7 08:06:45 2001


Query Started at 986655995.48
Query Ended at 986656005.22
Query Processed in 9.73 seconds


SQL statements processed: 1
Queries processed: 1

==================================================
```

qual11.v1

```
==================================================
Begin Execution at Sat Apr  7 08:06:45 2001


-- 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
85606.00	15053957.15
33354.00	14408297.40
154747.00	14407580.68
82865.00	14235489.78
76094.00	14094247.04
222.00	13937777.74
121271.00	13908336.00
55221.00	13716120.47
22819.00	13666434.28
76281.00	13646853.68
85298.00	13581154.93
85158.00	13554904.00
139684.00	13535538.72
31034.00	13498025.25
87305.00	13482847.04
10181.00	13445148.75
62323.00	13411824.30
26489.00	13377256.38
96493.00	13339057.83
56548.00	13329014.97
55576.00	13306843.35
159751.00	13306614.48
92406.00	13287414.50
182636.00	13223726.74
199969.00	13135288.21
62865.00	13001926.94
7284.00	12945298.19
197867.00	12944510.52
11562.00	12931575.51
75165.00	12916918.12
97175.00	12911283.50
140840.00	12896562.23
65241.00	12890600.46
166120.00	12876927.22
9035.00	12863828.70
144616.00	12853549.30
176723.00	12832309.74
170884.00	12792136.58
29790.00	12723300.33
95213.00	12555483.73
183873.00	12550533.05
171235.00	12476538.30
21533.00	12437821.32
17290.00	12432159.50
156397.00	12260623.50
122611.00	12222812.98
139155.00	12220319.25
146316.00	12215800.61
171381.00	12199734.52
198633.00	12078226.95
167417.00	12046637.62
59512.00	12043468.76
31688.00	12034893.64
159586.00	12001505.84
8993.00	11963814.30
120302.00	11857707.55
43536.00	11779340.52
9552.00	11776909.16
86223.00	11772205.08
53776.00	11758669.65
131285.00	11616953.74
91628.00	11611114.83
169644.00	11567959.72
182299.00	11567462.05
33107.00	11453818.76
104184.00	11436657.44
67027.00	11419127.14
176869.00	11371451.71
30885.00	11369674.79
54420.00	11345076.88
72240.00	11313951.05
178708.00	11294635.17
81298.00	11273686.13
158324.00	11243442.72
117095.00	11242535.24
176793.00	11237733.38
86091.00	11177793.79
116033.00	11145434.36
129058.00	11119112.20
193714.00	11104706.39
117195.00	11077217.96
49851.00	11043701.78
19791.00	11030662.62
75800.00	11012401.62
161562.00	10996371.69
10119.00	10980015.75
39185.00	10970042.56
47223.00	10950022.13
175594.00	10942923.05
111295.00	10893675.61
155446.00	10852764.57
156391.00	10839810.38
40884.00	10837234.19
141288.00	10837130.21
152388.00	10830977.82
33449.00	10830858.72
149035.00	10826130.02
162620.00	10814275.68
118324.00	10791788.10
38932.00	10777541.75
121294.00	10764225.22
48721.00	10762582.49
63342.00	10740132.60
5614.00	10724668.80
62266.00	10711143.10
100202.00	10696675.55
197741.00	10688560.72
169178.00	10648522.80
5271.00	10639392.65
34499.00	10584177.10
71108.00	10569117.56
137132.00	10539880.47
78451.00	10524873.24
150827.00	10503810.48
107237.00	10488030.84
101727.00	10473558.10
58708.00	10466280.44
89768.00	10465477.22
146493.00	10444291.58
55424.00	10444006.48
16560.00	10425574.74
133114.00	10415097.90
195810.00	10413625.20
76673.00	10391977.18
97305.00	10390890.57
134210.00	10387210.02
188536.00	10386529.92
122255.00	10335760.32
2682.00	10312966.10
43814.00	10303086.61
34767.00	10290405.18
165584.00	10273705.89
2231.00	10270415.55
111259.00	10263256.56
195578.00	10239795.82
21093.00	10217531.30
29856.00	10216932.54
133686.00	10213345.76
87745.00	10185509.40
135153.00	10179379.70
11773.00	10167410.84
76316.00	10165151.70
123076.00	10161225.78
91894.00	10130462.19
39741.00	10128387.52
111753.00	10119780.98

```
142729.00               10104748.89
116775.00               10097750.42
102589.00               10034784.36
186268.00               10012181.57
44545.00                10000286.48
23307.00                9966577.50
124281.00               9930018.90
69604.00                9925730.64
21971.00                9908982.03
58148.00                9895894.40
16532.00                9886529.90
159180.00               9883744.43
74733.00                9877582.88
35173.00                9858275.92
7116.00                 9856881.02
124620.00               9838589.14
122108.00               9829949.35
67200.00                9828690.69
164775.00               9821424.44
9039.00                 9816447.72
14912.00                9803102.20
190906.00               9791315.70
130398.00               9781674.27
119310.00               9776927.21
10132.00                9770930.78
107211.00               9757586.25
113958.00               9757065.50
37009.00                9748362.69
66746.00                9743528.76
134486.00               9731922.00
15945.00                9731096.45
55307.00                9717745.80
56362.00                9714922.83
57726.00                9711792.10
57256.00                9708621.00
112292.00               9701653.08
87514.00                9699492.53
174206.00               9680562.02
72865.00                9679043.34
114357.00               9671017.44
112807.00               9665019.21
115203.00               9661018.73
177454.00               9658906.35
161275.00               9634313.71
61893.00                9617095.44

... rows truncated ...

1048 rows processed.
Statement Processed in 21.02 seconds.

Ended Executing this Query at Sat Apr  7 08:07:06 2001


Query Started at 986656005.49
Query Ended at 986656026.52
Query Processed in 21.02 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual12.v1

```
====================================================
Begin Execution at Sat Apr  7 08:07:06 2001


-- using default substitutions

select
        l_shipmode,
        sum(case
                when o_orderpriority = '1-URGENT'
                        or o_orderpriority = '2-HIGH'
                        then 1
                else 0
        end) as high_line_count,
        sum(case
                when o_orderpriority <> '1-URGENT'
                        and o_orderpriority <> '2-HIGH'
                        then 1
                else 0
        end) as low_line_count
from
        orders,
        lineitem
where
        o_orderkey = l_orderkey
        and l_shipmode in ('MAIL', 'SHIP')
        and l_commitdate < l_receiptdate
        and l_shipdate < l_commitdate
and l_receiptdate >= to_date( '1994-01-01',  'YYYY-MM-
DD')
and l_receiptdate < add_months(to_date ('1994-01-01',
'YYYY-MM-DD'),  12)
group by
        l_shipmode
order by
        l_shipmode

L_SHIPMODE HIGH_LINE_COUNT        LOW_LINE_COUNT
MAIL       6202.00                9324.00
SHIP       6200.00                9262.00


2 rows processed.
Statement Processed in 8.12 seconds.

Ended Executing this Query at Sat Apr  7 08:07:14 2001


Query Started at 986656026.79
Query Ended at 986656034.91
Query Processed in 8.12 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual13.v1

```
====================================================
Begin Execution at Sat Apr  7 08:07:15 2001


-- using default substitutions

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

C_COUNT                CUSTDIST
0.00                   50004.00
9.00                   6641.00
10.00                  6566.00
11.00                  6058.00
8.00                   5949.00
12.00                  5553.00
13.00                  4989.00
19.00                  4748.00
7.00                   4707.00
18.00                  4625.00
15.00                  4552.00
17.00                  4530.00
14.00                  4484.00
20.00                  4461.00
16.00                  4323.00
21.00                  4217.00
22.00                  3730.00
6.00                   3334.00
23.00                  3129.00
24.00                  2622.00
25.00                  2079.00
5.00                   1972.00
26.00                  1593.00
27.00                  1185.00
4.00                   1033.00
28.00                  869.00
29.00                  559.00
3.00                   398.00
```

```
30.00                   373.00
31.00                   235.00
2.00                    144.00
32.00                   128.00
33.00                   71.00
34.00                   48.00
35.00                   33.00
1.00                    23.00
36.00                   17.00
37.00                   7.00
40.00                   4.00
38.00                   4.00
39.00                   2.00
41.00                   1.00


42 rows processed.
Statement Processed in 7.73 seconds.

Ended Executing this Query at Sat Apr  7 08:07:22 2001


Query Started at 986656035.17
Query Ended at 986656042.90
Query Processed in 7.73 seconds


SQL statements processed: 1
Queries processed: 1
```

==========

qual14.v1

==========

```
Begin Execution at Sat Apr  7 08:07:23 2001


-- using default substitutions

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

PROMO_REVENUE
16.38


1 row
```

==========

qual15.v1

==========

```
Begin Execution at Sat Apr  7 08:07:24 2001


-- using default substitutions

create view revenue0 (supplier_no, total_revenue) as
select
l_suppkey,
sum(l_extendedprice * (1 - l_discount))
from
lineitem
where
l_shipdate >= to_date( '1996-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1996-01-01',
'YYYY-MM-DD'), 3)
group by
l_suppkey
Statement Processed in 0.04 seconds.

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

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


1 row
```

==========

qual16.v1

==========

```
Begin Execution at Sat Apr  7 08:07:35 2001


-- using default substitutions

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

P_BRAND    P_TYPE                      P_SIZE
SUPPLIER_CNT
Brand#41   MEDIUM BRUSHED TIN         3.00
28.00
Brand#54   STANDARD BRUSHED COPPER    14.00
27.00
Brand#11   STANDARD BRUSHED TIN       23.00
24.00
Brand#11   STANDARD BURNISHED BRASS   36.00
24.00
Brand#15   MEDIUM ANODIZED NICKEL     3.00
24.00
Brand#15   SMALL ANODIZED BRASS       45.00
24.00
Brand#15   SMALL BURNISHED NICKEL     19.00
24.00
Brand#21   MEDIUM ANODIZED COPPER     3.00
24.00
Brand#22   SMALL BRUSHED NICKEL       3.00
24.00
Brand#22   SMALL BURNISHED BRASS      19.00
24.00
Brand#25   MEDIUM BURNISHED COPPER    36.00
24.00
Brand#31   PROMO POLISHED COPPER      36.00
```

```
24.00
Brand#33   LARGE POLISHED TIN        23.00
24.00
Brand#33   PROMO POLISHED STEEL      14.00
24.00
Brand#35   PROMO BRUSHED NICKEL      14.00
24.00
Brand#41   ECONOMY BRUSHED STEEL     9.00
24.00
Brand#41   ECONOMY POLISHED TIN      19.00
24.00
Brand#41   LARGE PLATED COPPER       36.00
24.00
Brand#42   ECONOMY PLATED BRASS      3.00
24.00
Brand#42   STANDARD POLISHED TIN     49.00
24.00
Brand#43   PROMO BRUSHED TIN         3.00
24.00
Brand#43   SMALL ANODIZED COPPER     36.00
24.00
Brand#44   STANDARD POLISHED NICKEL  3.00
24.00
Brand#52   ECONOMY PLATED TIN        14.00
24.00
Brand#52   STANDARD BURNISHED NICKEL 3.00
24.00
Brand#53   MEDIUM ANODIZED STEEL     14.00
24.00
Brand#14   PROMO ANODIZED NICKEL     45.00
23.00
Brand#32   ECONOMY PLATED BRASS      9.00
23.00
Brand#52   SMALL ANODIZED COPPER     3.00
23.00
Brand#11   ECONOMY BRUSHED COPPER    45.00
20.00
Brand#11   ECONOMY PLATED BRASS      23.00
20.00
Brand#11   LARGE BRUSHED COPPER      49.00
20.00
Brand#11   LARGE POLISHED COPPER     49.00
20.00
Brand#12   STANDARD ANODIZED TIN     49.00
20.00
Brand#12   STANDARD PLATED BRASS     19.00
20.00
Brand#13   ECONOMY BRUSHED BRASS     9.00
20.00
Brand#13   ECONOMY BURNISHED STEEL   14.00
20.00
Brand#13   LARGE BURNISHED NICKEL    19.00
20.00
Brand#13   MEDIUM BURNISHED COPPER   36.00
20.00
Brand#13   SMALL BRUSHED TIN         45.00
20.00
Brand#13   STANDARD ANODIZED COPPER  3.00
20.00
Brand#13   STANDARD PLATED NICKEL    23.00
20.00
Brand#14   ECONOMY ANODIZED COPPER   14.00
20.00
Brand#14   ECONOMY PLATED TIN        36.00
20.00
Brand#14   ECONOMY POLISHED NICKEL   3.00
20.00
Brand#14   MEDIUM ANODIZED NICKEL    3.00
20.00
Brand#14   SMALL POLISHED TIN        14.00
20.00
Brand#15   MEDIUM ANODIZED COPPER    9.00
20.00
Brand#15   MEDIUM PLATED TIN         23.00
20.00
Brand#15   PROMO PLATED BRASS        14.00
20.00
Brand#15   SMALL ANODIZED COPPER     45.00
20.00
Brand#15   SMALL PLATED COPPER       49.00
20.00
Brand#15   STANDARD PLATED TIN       3.00
20.00
Brand#21   LARGE ANODIZED COPPER     36.00
20.00
Brand#21   LARGE BRUSHED TIN         3.00
20.00
Brand#21   MEDIUM ANODIZED COPPER    14.00
20.00
Brand#21   PROMO BRUSHED TIN         36.00
20.00
Brand#21   PROMO POLISHED NICKEL     45.00
20.00
Brand#21   SMALL ANODIZED COPPER     9.00
20.00
Brand#21   SMALL POLISHED NICKEL     23.00
20.00
Brand#22   LARGE ANODIZED COPPER     36.00
20.00
Brand#22   LARGE BRUSHED COPPER      49.00
20.00
Brand#22   PROMO ANODIZED TIN        49.00
20.00
Brand#22   PROMO POLISHED BRASS      45.00
20.00
Brand#22   SMALL BURNISHED STEEL     45.00
20.00
Brand#23   MEDIUM ANODIZED STEEL     45.00
20.00
Brand#23   PROMO POLISHED STEEL      23.00
20.00
Brand#23   STANDARD BRUSHED TIN      14.00
20.00
Brand#23   STANDARD PLATED NICKEL    36.00
20.00
Brand#24   PROMO PLATED COPPER       49.00
20.00
Brand#24   PROMO PLATED STEEL        49.00
20.00
Brand#24   PROMO POLISHED STEEL      9.00
20.00
Brand#24   STANDARD BRUSHED TIN      36.00
20.00
Brand#25   LARGE ANODIZED BRASS      3.00
20.00
Brand#25   PROMO BURNISHED TIN       3.00
20.00
Brand#31   ECONOMY POLISHED NICKEL   3.00
20.00
Brand#31   MEDIUM PLATED TIN         45.00
20.00
Brand#31   SMALL ANODIZED STEEL      14.00
20.00
Brand#32   ECONOMY ANODIZED COPPER   36.00
20.00
Brand#32   ECONOMY BRUSHED NICKEL    49.00
20.00
Brand#32   LARGE ANODIZED TIN        19.00
20.00
Brand#32   MEDIUM BURNISHED COPPER   19.00
20.00
Brand#32   SMALL ANODIZED STEEL      45.00
20.00
Brand#33   ECONOMY POLISHED COPPER   19.00
20.00
Brand#33   PROMO PLATED NICKEL       14.00
20.00
Brand#33   SMALL POLISHED TIN        9.00
20.00
Brand#33   STANDARD ANODIZED BRASS   49.00
20.00
Brand#33   STANDARD BURNISHED BRASS  45.00
20.00
Brand#34   ECONOMY BRUSHED NICKEL    49.00
20.00
Brand#34   LARGE BRUSHED BRASS       19.00
20.00
Brand#34   SMALL BRUSHED TIN         3.00
20.00
Brand#34   STANDARD PLATED COPPER    9.00
20.00
Brand#35   LARGE ANODIZED NICKEL     3.00
20.00
Brand#35   MEDIUM ANODIZED BRASS     45.00
20.00
Brand#35   MEDIUM ANODIZED STEEL     23.00
20.00
Brand#35   PROMO ANODIZED COPPER     49.00
20.00
Brand#35   SMALL POLISHED COPPER     14.00
20.00
Brand#41   LARGE ANODIZED STEEL      3.00
20.00
Brand#41   LARGE BRUSHED NICKEL      23.00
20.00
Brand#41   LARGE BURNISHED COPPER    3.00
20.00
Brand#41   MEDIUM PLATED STEEL       19.00
20.00
Brand#41   SMALL BURNISHED COPPER    23.00
```

```
20.00
Brand#42   MEDIUM BURNISHED BRASS    14.00
20.00
Brand#42   SMALL BURNISHED COPPER    3.00
20.00
Brand#43   ECONOMY POLISHED COPPER   9.00
20.00
Brand#43   SMALL PLATED STEEL        3.00
20.00
Brand#43   STANDARD BURNISHED TIN    23.00
20.00
Brand#44   LARGE ANODIZED STEEL      23.00
20.00
Brand#44   PROMO ANODIZED TIN        23.00
20.00
Brand#51   ECONOMY BRUSHED BRASS     49.00
20.00
Brand#51   ECONOMY POLISHED NICKEL   9.00
20.00
Brand#51   MEDIUM BRUSHED TIN        9.00
20.00
Brand#51   MEDIUM PLATED BRASS       9.00
20.00
Brand#51   PROMO BURNISHED BRASS     9.00
20.00
Brand#51   SMALL PLATED NICKEL       49.00
20.00
Brand#51   STANDARD ANODIZED NICKEL  49.00
20.00
Brand#51   STANDARD BRUSHED COPPER   3.00
20.00
Brand#52   ECONOMY ANODIZED BRASS    3.00
20.00
Brand#52   ECONOMY BRUSHED COPPER    49.00
20.00
Brand#52   LARGE ANODIZED NICKEL     45.00
20.00
Brand#52   MEDIUM ANODIZED TIN       23.00
20.00
Brand#52   MEDIUM BURNISHED TIN      45.00
20.00
Brand#52   SMALL PLATED COPPER       36.00
20.00
Brand#52   STANDARD ANODIZED BRASS   45.00
20.00
Brand#53   ECONOMY PLATED COPPER     45.00
20.00
Brand#53   PROMO ANODIZED COPPER     49.00
20.00
Brand#53   PROMO BRUSHED COPPER      23.00
20.00
Brand#53   PROMO PLATED TIN          19.00
20.00
Brand#53   PROMO POLISHED NICKEL     3.00
20.00
Brand#53   SMALL ANODIZED STEEL      9.00
20.00
Brand#53   SMALL BRUSHED COPPER      3.00
20.00
Brand#53   SMALL BRUSHED NICKEL      3.00
20.00
Brand#54   ECONOMY PLATED STEEL      9.00
20.00
Brand#54   ECONOMY POLISHED TIN      3.00
20.00
Brand#54   SMALL BRUSHED BRASS       19.00
20.00
Brand#55   MEDIUM ANODIZED COPPER    3.00
20.00
Brand#55   PROMO BURNISHED STEEL     14.00
20.00
Brand#55   PROMO POLISHED NICKEL     49.00
20.00
Brand#55   STANDARD ANODIZED BRASS   19.00
20.00
Brand#55   STANDARD BURNISHED COPPER 45.00
20.00
Brand#43   ECONOMY ANODIZED TIN      3.00
19.00
Brand#11   ECONOMY ANODIZED BRASS    14.00
16.00
Brand#11   ECONOMY ANODIZED BRASS    23.00
16.00
Brand#11   ECONOMY ANODIZED COPPER   14.00
16.00
Brand#11   ECONOMY BRUSHED BRASS     49.00
16.00
Brand#11   ECONOMY BRUSHED STEEL     19.00
16.00
Brand#11   ECONOMY BURNISHED NICKEL  23.00
16.00
Brand#11   LARGE ANODIZED COPPER     14.00
16.00
Brand#11   LARGE BRUSHED TIN         45.00
16.00
Brand#11   LARGE BURNISHED COPPER    23.00
16.00
Brand#11   LARGE BURNISHED NICKEL    36.00
16.00
Brand#11   LARGE PLATED STEEL        14.00
16.00
Brand#11   MEDIUM BRUSHED NICKEL     14.00
16.00
Brand#11   MEDIUM BRUSHED STEEL      49.00
16.00
Brand#11   MEDIUM BURNISHED NICKEL   49.00
16.00
Brand#11   MEDIUM BURNISHED TIN      3.00
16.00
Brand#11   MEDIUM PLATED COPPER      9.00
16.00
Brand#11   PROMO ANODIZED BRASS      19.00
16.00
Brand#11   PROMO ANODIZED BRASS      49.00
16.00
Brand#11   PROMO ANODIZED STEEL      45.00
16.00
Brand#11   PROMO PLATED BRASS        45.00
16.00
Brand#11   SMALL ANODIZED TIN        45.00
16.00
Brand#11   SMALL BRUSHED STEEL       49.00
16.00
Brand#11   SMALL BURNISHED COPPER    19.00
16.00
Brand#11   SMALL BURNISHED COPPER    45.00
16.00
Brand#11   SMALL BURNISHED NICKEL    14.00
16.00
Brand#11   SMALL POLISHED NICKEL     36.00
16.00
Brand#11   STANDARD ANODIZED BRASS   19.00
16.00
Brand#11   STANDARD ANODIZED COPPER  14.00
16.00
Brand#11   STANDARD BRUSHED STEEL    45.00
16.00
Brand#11   STANDARD POLISHED NICKEL  23.00
16.00
Brand#12   ECONOMY ANODIZED TIN      14.00
16.00
Brand#12   ECONOMY BRUSHED COPPER    9.00
16.00
Brand#12   ECONOMY BRUSHED COPPER    36.00
16.00
Brand#12   ECONOMY BURNISHED BRASS   9.00
16.00
Brand#12   ECONOMY BURNISHED NICKEL  36.00
16.00
Brand#12   LARGE ANODIZED BRASS      14.00
16.00
Brand#12   LARGE ANODIZED COPPER     9.00
16.00
Brand#12   LARGE ANODIZED STEEL      23.00
16.00
Brand#12   LARGE BURNISHED TIN       36.00
16.00
Brand#12   LARGE PLATED COPPER       49.00
16.00
Brand#12   LARGE POLISHED COPPER     49.00
16.00
Brand#12   MEDIUM PLATED COPPER      19.00
16.00
Brand#12   MEDIUM PLATED NICKEL      23.00
16.00
Brand#12   PROMO ANODIZED BRASS      45.00
16.00
Brand#12   PROMO ANODIZED STEEL      49.00
16.00
Brand#12   PROMO BURNISHED STEEL     9.00
16.00
Brand#12   SMALL BRUSHED NICKEL      36.00
16.00
Brand#12   SMALL BRUSHED TIN         45.00
16.00
Brand#12   STANDARD ANODIZED BRASS   3.00
16.00
Brand#12   STANDARD ANODIZED NICKEL  14.00
16.00
Brand#12   STANDARD BRUSHED BRASS    3.00
```

```
16.00
Brand#12   STANDARD BRUSHED TIN      9.00
16.00
Brand#12   STANDARD BRUSHED TIN      36.00
16.00
Brand#12   STANDARD POLISHED COPPER  9.00
16.00
Brand#13   ECONOMY ANODIZED STEEL    45.00
16.00
Brand#13   ECONOMY POLISHED BRASS    3.00
16.00
Brand#13   LARGE BRUSHED NICKEL      23.00
16.00
Brand#13   LARGE BURNISHED NICKEL    9.00
16.00
Brand#13   MEDIUM BRUSHED STEEL      49.00
16.00

... rows truncated ...


18314 rows processed.
Statement Processed in 3.51 seconds.

Ended Executing this Query at Sat Apr  7 08:07:38 2001


Query Started at 986656055.31
Query Ended at 986656058.82
Query Processed in 3.51 seconds


SQL statements processed: 1
Queries processed: 1

=====================================================
```

qual17.v1

```
=====================================================
Begin Execution at Sat Apr  7 08:07:39 2001


-- using default substitutions

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

AVG_YEARLY
348406.05


1 row

=====================================================
```

qual18.v1

```
=====================================================
Begin Execution at Sat Apr  7 08:07:55 2001


-- using default substitutions

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

C_NAME                    C_CUSTKEY
O_ORDERKEY              O_ORDERDATE
O_TOTALPRICE            SUM(L_QUANTITY)
Customer#000128120        128120.00
4722021.00              1994-04-07
544089.09               323.00
Customer#000144617        144617.00
3043270.00              1997-02-12
530604.44               317.00
Customer#000013940        13940.00
2232932.00              1997-04-13
522720.61               304.00
Customer#000066790        66790.00
2199712.00              1996-09-30
515531.82               327.00
Customer#000046435        46435.00
4745607.00              1997-07-03
508047.99               309.00
Customer#000015272        15272.00
3883783.00              1993-07-28
500241.33               302.00
Customer#000146608        146608.00
3342468.00              1994-06-12
499794.58               303.00
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
Customer#000107590        107590.00
4267751.00              1994-11-04
485141.38               301.00
Customer#000050008        50008.00
2366755.00              1996-12-09
483891.26               302.00
Customer#000015619        15619.00
3767271.00              1996-08-07
480083.96               318.00
Customer#000077260        77260.00
1436544.00              1992-09-12
479499.43               307.00
Customer#000109379        109379.00
5746311.00              1996-10-10
478064.11               302.00
Customer#000054602        54602.00
5832321.00              1997-02-09
471220.08               307.00
Customer#000105995        105995.00
2096705.00              1994-07-03
469692.58               307.00
Customer#000148885        148885.00
2942469.00              1992-05-31
469630.44               313.00
Customer#000114586        114586.00
551136.00               1993-05-19
469605.59               308.00
Customer#000105260        105260.00
5296167.00              1996-09-06
469360.57               303.00
Customer#000147197        147197.00
1263015.00              1997-02-02
467149.67               320.00
Customer#000064483        64483.00
```

```
2745894.00              1996-07-04
466991.35               304.00
Customer#000136573         136573.00
2761378.00              1996-05-31
461282.73               301.00
Customer#000016384         16384.00
502886.00               1994-04-12
458378.92               312.00
Customer#000117919         117919.00
2869152.00              1996-06-20
456815.92               317.00
Customer#000012251         12251.00
735366.00               1993-11-24
455107.26               309.00
Customer#000120098         120098.00
1971680.00              1995-06-14
453451.23               308.00
Customer#000066098         66098.00
5007490.00              1992-08-07
453436.16               304.00
Customer#000117076         117076.00
4290656.00              1997-02-05
449545.85               301.00
Customer#000129379         129379.00
4720454.00              1997-06-07
448665.79               303.00
Customer#000126865         126865.00
4702759.00              1994-11-07
447606.65               320.00
Customer#000088876         88876.00
983201.00               1993-12-30
446717.46               304.00
Customer#000036619         36619.00
4806726.00              1995-01-17
446704.09               328.00
Customer#000141823         141823.00
2806245.00              1996-12-29
446269.12               310.00
Customer#000053029         53029.00
2662214.00              1993-08-13
446144.49               302.00
Customer#000018188         18188.00
3037414.00              1995-01-25
443807.22               308.00
Customer#000066533         66533.00
29158.00                1995-10-21
443576.50               305.00
Customer#000037729         37729.00
4134341.00              1995-06-29
441082.97               309.00
Customer#000003566         3566.00
2329187.00              1998-01-04
439803.36               304.00
Customer#000045538         45538.00
4527553.00              1994-05-22
436275.31               305.00
Customer#000081581         81581.00
4739650.00              1995-11-04
435405.90               305.00
Customer#000119989         119989.00
1544643.00              1997-09-20
434568.25               320.00
Customer#000003680         3680.00
3861123.00              1998-07-03
433525.97               301.00
Customer#000113131         113131.00
967334.00               1995-12-15
432957.75               301.00
Customer#000141098         141098.00
565574.00               1995-09-24
430986.69               301.00
Customer#000093392         93392.00
5200102.00              1997-01-22
425487.51               304.00
Customer#000015631         15631.00
1845057.00              1994-05-12
419879.59               302.00
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


57 rows processed.
Statement Processed in 94.96 seconds.

Ended Executing this Query at Sat Apr  7 08:09:30 2001


Query Started at 986656075.76
Query Ended at 986656170.72
Query Processed in 94.96 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual19.v1

```
====================================================
Begin Execution at Sat Apr  7 08:09:30 2001


-- using default substitutions

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

REVENUE
3083843.06


1 row
```

```
====================================================
```

qual20.v1

```
====================================================
Begin Execution at Sat Apr  7 08:09:40 2001


-- using default substitutions

select
s_name,
s_address
from
supplier,
nation
where
s_suppkey in (
select
ps_suppkey
from
partsupp
where
ps_partkey in (
select
p_partkey
from
part
where
p_name like 'forest%'
)
and ps_availqty > (
select
0.5 * sum(l_quantity)
from
lineitem
where
l_partkey = ps_partkey
and l_suppkey = ps_suppkey
and l_shipdate >= to_date ('1994-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1994-01-01',
'YYYY-MM-DD'), 12)
)
)
and s_nationkey = n_nationkey
and n_name = 'CANADA'
order by
s_name

S_NAME                    S_ADDRESS
Supplier#000000020        iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197
YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226        83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285
Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378        FfbhyCxWvcPrO8ltp9
Supplier#000000402        i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530        0qwCMwobKY
OcmLyfRXlagA8ukENJv,
Supplier#000000688        D
fw5ocppmZpYBBIPI718hCihLDZ5KhKX
Supplier#000000710        f19YPvOyb
QoYwjKC,oPycpGfieBAcwKJo
Supplier#000000736
l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL7
Supplier#000000761
zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
Supplier#000000884        bmhEShejaS
Supplier#000000887        urEaTejH5POADP2ARrf
Supplier#000000935        ij98czM
2KzWe7dDTOxB8sq0UfCdvrX
Supplier#000000975        ,AC e,tBpNwKb5xMUzeohxlRn,
hdZJo73gFQF8y
Supplier#000001263        rQWr6nf8ZhB2TAiIDIvo5Io
Supplier#000001399        LmrocnIMSyYOWuANx7
Supplier#000001446        lch9HMNU1R7a0LIybsUodVknk6
Supplier#000001454        TOpimgu2TVXIjhiL93h,
Supplier#000001500        wDmF5xLxtQch9ctVu,
Supplier#000001602        uKNWIeafaM644
Supplier#000001626        UhxNRzUu1dtFmp0
Supplier#000001682        pXTkGxrTQVyH1Rr
Supplier#000001699        Q9C4rfJ26oijVPqqcqVXeRI
Supplier#000001700        7hMlCof1Y5zLFg
Supplier#000001726        TeRY7TtTH24sEword7yAaSkjx8
Supplier#000001730        Rc8e,1Pybn r6zo0VJIEiD0UD
vhk
Supplier#000001746
qWsendlOekQG1aW4uq06uQaCm51se8lirv7 hBRd
Supplier#000001752        Fra7outx41THYJaRThdOGiBk
Supplier#000001856
jXcRgzYF0ah05iR8p6w5SbJJLcUGyYiURPvFwUWM
Supplier#000001931        FpJbMU2h6ZR2eBv8I9NIxF
Supplier#000001939        Nrk,JA4bfReUs
Supplier#000001990
DSDJkCgBJzuPg1yuM,CUdLnsRliOxkkHezTCA
Supplier#000002020        jB6r1d7MxP6co
Supplier#000002022         dwebGX7Id2pc25YvY33
Supplier#000002036        20ytTtVObjKUUI2WCB0A
Supplier#000002204
uYmlr46C06udCqanj0KiRsoTQakZsEyssL
Supplier#000002243        nSOEV3JeOU79
Supplier#000002245
hz2qWXWVjOyKhqPYMoEwz6zFkrTaDM
Supplier#000002282
ES21K9dxoW1I1TzWCj7ekdlNwSWnv1Z  6mQ,BKn
Supplier#000002303        nCoWfpB6YOymbgOht7ltfklpkHl
Supplier#000002373        RzHSxOTQmElCjxIBiVA52Z
JB58rJhPRylR
Supplier#000002419        qydBQd14I5l5mVXa4fYY
Supplier#000002481
nLKHUOn2Ml9TOA06Znq9GEMcIlMO2
Supplier#000002571        JZUugz04c iJFLrlGsz9O N,W
1rVHNIReyq
Supplier#000002585
CsPoKpw2QuTY4AV1NkWuttneIa4SN
Supplier#000002630        ZIQAvjNUY9KH5ive zm7k
VlPiDl7CCo21
Supplier#000002719
4nnzQI2CbqREQUuIsXTBVUkaP4mNS3
Supplier#000002721        HVdFAN2JHMQSpKm
Supplier#000002730        lIFxR4fzm31C6,muzJwl84z
Supplier#000002775        yDclaDaBD4ihH
Supplier#000002853        rTNAOItXka
Supplier#000002875        6JgMi
9Qt6VmwL3Ltt1SRlKww0keLQ,RAZa
Supplier#000002934        m,trBENywSArwg3DhB
Supplier#000002941        Naddba 8YTEKekZyP0
Supplier#000002960
KCPCEsRGGo6vx8TygHh60nAYf9rStQT2T
Supplier#000002980
B9k9yVsyaXvWktOSHezqHiAEp9id0SKzkw
Supplier#000003062        LSQNgqY1xnOzz9zBCapy7HwOZQ
Supplier#000003087        ANwe8QsZ4rgj1HSqVz991eWQ
Supplier#000003089        s5b VCIZqMSZVa r
g7LTdcg29GbTE7rI1x
Supplier#000003095        HxON3jJhUi3zjt,r mTD
Supplier#000003201
E87yws6I,t0qNs4QW7UzExKiJnJDZWue
Supplier#000003213        pxrRP4irQ1VoyfQ,dTf3
Supplier#000003241        j06SU,LS9O3mwjAMOViANeIhb
Supplier#000003275        9xO4nyJ2QJcX6vGf
Supplier#000003288
EDdfNt7E5Uc,xLTupoIgYL4yY7ujh,
Supplier#000003313
El2I7we,049SPrvomUm4hZwJoOhZkvLxLJXgVH
Supplier#000003314
jnisU8MzqO4iUB3zsPcrysMw3DDUojS4q7LD
Supplier#000003380
jPv0V,pszouuFT3YsAqlP,kxT3u,gTFiEbRt,x
Supplier#000003403        e3X2o ,KCG9tsHji8A
XXCxiF2hZWBw
Supplier#000003421        Sh3dt9W5oeofFWovnFhrg,
Supplier#000003441        zvFJIzS,oUuShHjpcX
Supplier#000003590        sy79CMLxqb,Cbo
Supplier#000003607        lNqFHQYjwSAkf
Supplier#000003625
qY588W0Yk5iaUy1RXTgNrEKrMAjBYHcKs
Supplier#000003656        eEYmmO2gmD JdfG32XtDgJV,db56
Supplier#000003782
iVsPZg7bk06TqNMwi0LKbLUrC1zmrg
Supplier#000003918        meRvRCsJoAbfqd0Re4
Supplier#000003941        Pmb05mQfBMS618O7WKqZJ 9vyv
Supplier#000003994        W00LZp3NjK0
Supplier#000004005        V723F1wCy2eA4OgIu8TjBtOVUHp
Supplier#000004033        ncsAhv9Je,kFXTNjfb2
Supplier#000004140        0hL7DJyYjcHL
Supplier#000004165
wTJ2dZNQA8P2oi99N6DT47ndHy,XKD2
Supplier#000004207        tF64pwiOM4IkWjN3mS,e06WuAjLx
Supplier#000004236
dl,HPtJmGipxYsSqn9wmqkuWjst,mCeJ8O6T
Supplier#000004246        Xha aXQF7u4qU3LsHD
Supplier#000004278        bBddbpBxIVp Di9
Supplier#000004343        GK3sbopqrQEkWLMvVBFCG
Supplier#000004346        S3076LEOwo
Supplier#000004388        VfZ l1J,mwp4aS
Supplier#000004406
Ah0ZaLu6VwufPWUz,7kbXgYZhauEaHqGIg
```

```
Supplier#000004430
yvSsKNSTL5HLXBET4luOsPNLxKzAMk
Supplier#000004522
xXtCKwsZDArxIBGDfzX2PgobGZsBg
Supplier#000004527        p pVXCnxgcklWF6A1o3OHY3qW6
Supplier#000004542        NJSbLJDroYG2y1r3rDiKg
Supplier#000004574        1HvGwnVueZ5CIndc
Supplier#000004655        67NqBc4 t3PG3F8aO
IsqWNq4kGaPowYL
Supplier#000004701        6jX4u47URzIMHf
Supplier#000004711        bEzjp1QdQu ls2ERMxv0km
vn6bu2zXlL1
Supplier#000004987         UFx1upJ8MvOvgFjA8
Supplier#000005000        DeX804 w0H8FrCUvahgy
ilbuzBX3NK
Supplier#000005100        OfvYPs3Io,wEvvLHNaLuCX
Supplier#000005192        JDp4rhXiDw0kf6RH
Supplier#000005195        Woi3b2ZaicPh ZSfu1EfXhE
Supplier#000005283        5fxYXxwXy,TQX,MqDC2hxzyQ
Supplier#000005300        gXG28YqpxU
Supplier#000005386        Ub6AAfHpWLWP
Supplier#000005426        9Dz2OVT1q
sb4BK71ljQ1XjPBYRPvO
Supplier#000005484        saFdOR
qW7AFY,3asPqiiAa11Mo22pCoN0BtPrKo
Supplier#000005505        d2sbjG43KwMPX
Supplier#000005506        On f5ypzoWgB
Supplier#000005516
XsN99Ks9wEvcohU6jRD2MeebQFf76mD8vovuY
Supplier#000005536
Nzo9tGkpgbHT,EZ4D,77MYKl4ah1C
Supplier#000005605        7Vj6Eil0mThqkM
Supplier#000005631        14TVrjlzo2SJEBYCDgpMwTlvwSqC
Supplier#000005730        5rkb0PSews
HvxkL8JaD41UpnSF2cg8H1
Supplier#000005736        2dq XTYhtYWSfp
Supplier#000005737        dmEWcS32C3kx,d,B95 OmYn48
Supplier#000005797        ,o,OebwRbSDmVl9gN9fpWPCiqB
UogvlSR
Supplier#000005836        tx3SjPD2ZuWGFBRH,
Supplier#000005875
lK,sYiGzB94hSyHy9xvSZFbVQNCZe2LXZuGbS
Supplier#000005974
REhR5jE,lLusQXvf54SwYySgsSSVFhu
Supplier#000005989        rjFY,5kgLpBu7c
Supplier#000006059        4m0cv8MwJ9yX2vlwI Z
Supplier#000006065        UiI2Cy3W4Tu5sLk
LuvXLRy6KihlGv
Supplier#000006070        TalC5m0pDrO6DZbngfmGmqe
Supplier#000006109        rY5gbfh3dKHnylcQUTPGCwnbe
Supplier#000006121        S92ycWwEzYYw4GspCBJN1WMuHhoZ
Supplier#000006215
j2iEbTsl,5PWdqWZ7k1yiISb7qtiiZljDIPEo
Supplier#000006217        RVN23SYT9jenUeaWGXUd
Supplier#000006274        S3yTZWqxTKUq g QQgcW9
AqhCkNZsW51hHuwU
Supplier#000006435        xIgE69XszYbnO4Eon7cHHO8y
Supplier#000006463        7 wkdj2EO49iotley2kmIM
ADpLSszGV3RNWj
Supplier#000006493        ojV
f,sNaB6Hm7r,fknDVTL63raJgAjZK
Supplier#000006521        b9 2zjHzxR
Supplier#000006607        3F 2e2gqD5u5B
Supplier#000006706
Ak4ga,ePu1QZ6C3qkrqjosaX0gxvqS9vkbe
Supplier#000006761
n4jhxGMqB5prD1HhpLvwrWStOLlla
Supplier#000006808        HGd2Xo 9nEcHJhZvXjXxWKIpApT
Supplier#000006858
fnlINT885vBBhsWwTGiZ0o22thwGY16h GHJj21
Supplier#000006872        XIDPiA7PLXCWK6SeEcld
Supplier#000006949        mLxYUJhsGcLtKe ,GFirNu183AvT
Supplier#000006985
PrUUiboQpy,OtgJ01Z4BxJQUyrw9c3I
Supplier#000007072        2tRyX9M1a
4Rcm57s779F1ANG9jlpK
Supplier#000007098
G3j8g0KC4OcbAu2OVoPHrXQWMCUdjq8wgCHOExu
Supplier#000007135        ls DoKV7V5ulfQy9V
Supplier#000007160
TqDGBULB3cTqIT6FKDvm9BS4e4v,zwYiQPb
Supplier#000007169        tEc95D2moN9S84nd55O,dlnW
Supplier#000007322        wr7dgte5q
MAjiY0uwmi3MyDkSMX1
Supplier#000007365        51xhROLvQMJ05DndtZWt
Supplier#000007398        V8eE6oZ00OFNU,
Supplier#000007402        4UVv58ery1rjmqSR5
Supplier#000007448        yhhpWiJi7EJ6Q5VCaQ
Supplier#000007477        9m9j0wfhWzCvVHxkU,PpAxwSH0h
Supplier#000007509        q8,V6LJRoHJjHcOuSG7aLTMg
Supplier#000007561        rMcFg2530VC
Supplier#000007789
rQ7cUcPrtudOyO3svNSkimqH6qrfWT2Sz
Supplier#000007801        69fi,U1r6enUb
Supplier#000007818        yhhc2CQec Jrvc8zqBi83
Supplier#000007885
u3sicchh5ZpyTUpN1cJKNcAoabIWgY
Supplier#000007918        r,v9mBQ6LoEYyj1
Supplier#000007926        ErzCF80K9Uy
Supplier#000007957        ELwnio14ssoU1 dRyZIL OK3Vtzb
Supplier#000007965        F7Un5lJ7p5hhj
Supplier#000007968
DsF9UlZ2Fo6HXN9aErvyg1ikHoD582HSGZpP
Supplier#000007998        LnASFBfYRFOo9d6d,asBvVq9Lo2P
Supplier#000008168        aOa82a8ZbKCnfDLX
Supplier#000008231        IK7eGw Yj90sTdpsP,vcqWxLB
Supplier#000008243        2AyePMkDqmzVzjGTizXthFLo8h
EiudCMxOmIIG
Supplier#000008275        BlbNDfWg,gpXKQlLN
Supplier#000008323        75I18sZmASwm
POeheRMdj9tmpyeQ,BfCXN5BIAb
Supplier#000008366
h778cEj14BuW9OEKlvPTWq4iwASR6EBBXN7zeS8
Supplier#000008423
RQhKnkAhR0DAr3Ix4Q1weMMn00hNe Kq
Supplier#000008480        4sSDA4ACReklNjEm5T6b
Supplier#000008532
Uc29q4,5xVdDOF87UZrxhr4xWS0ihEUXuh
Supplier#000008595        MH0iB73GQ3z UW3O DbCbqmc
Supplier#000008610
SgVgP90vP452sUNTgzL9zKwXHXAzV6tV
Supplier#000008705        aE,trRNdPx,4yinTD9O3DebDIp
Supplier#000008742        HmPlQEzKCPEcTUL14,kKq
Supplier#000008841        I 85Lu1sekbg2xrSIzm0
Supplier#000008895
2cH4okfaLSZTTg8sKRbbJQxkmeFu2Esj
Supplier#000008967        2kwEHyMG
7FwozNImAUE6mH0hYtqYculJM
Supplier#000008972        w2vF6 D5YZO3visPXsqVfLADTK
Supplier#000009032        qK,trB6Sdy4Dz1BRUFNy
Supplier#000009147        rOAuryHxpZ9eOvx
Supplier#000009252        F7cZaPUHwh1 ZKyj3xmAVWC1XdP
ue1p5m,i
Supplier#000009278        RqYTzgxj93CLX 0mcYfCENOefD
Supplier#000009327        uoqMdf7e7Gj9dbQ53
Supplier#000009430        igRqmneFt
Supplier#000009567        r4Wfx4c3xsEAjcGj71HHZByornl
D9vrztXlv4
Supplier#000009601        51m637bO,Rw5DnHWFUvLacRx9
Supplier#000009709        rRnCbHYgDgl9PZYnyWKVYSUW0vKg
Supplier#000009753        wLhVEcRmd7PkJF4FBnGK7Z
Supplier#000009796        z,y4Idmr15DOvPUqYG
Supplier#000009799         4wNjXGa4OKWl
Supplier#000009811        E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812
APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862        rJzweWeN58

... rows truncated ...


204 rows processed.
Statement Processed in 5.22 seconds.

Ended Executing this Query at Sat Apr  7 08:09:45 2001


Query Started at 986656180.68
Query Ended at 986656185.90
Query Processed in 5.22 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual21.v1

```
====================================================
Begin Execution at Sat Apr  7 08:09:46 2001


-- using default substitutions

select * from (
select
s_name,
count(*)  numwait
from
supplier,
```

```
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.l_receiptdate > l1.l_commitdate
and exists (
select
*
from
lineitem l2
where
l2.l_orderkey = l1.l_orderkey
and l2.l_suppkey <> l1.l_suppkey
)
and not exists (
select
*
from
lineitem l3
where
l3.l_orderkey = l1.l_orderkey
and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate
)
and s_nationkey = n_nationkey
and n_name = 'SAUDI ARABIA'
group by
s_name
order by
numwait desc,
s_name)
where rownum <= 100
```

S_NAME	NUMWAIT
Supplier#000002829	20.00
Supplier#000005808	18.00
Supplier#000000262	17.00
Supplier#000000496	17.00
Supplier#000002160	17.00
Supplier#000002301	17.00
Supplier#000002540	17.00
Supplier#000003063	17.00
Supplier#000005178	17.00
Supplier#000008331	17.00
Supplier#000002005	16.00
Supplier#000002095	16.00
Supplier#000005799	16.00
Supplier#000005842	16.00
Supplier#000006450	16.00
Supplier#000006939	16.00
Supplier#000009200	16.00
Supplier#000009727	16.00
Supplier#000000486	15.00
Supplier#000000565	15.00
Supplier#000001046	15.00
Supplier#000001047	15.00
Supplier#000001161	15.00
Supplier#000001336	15.00
Supplier#000001435	15.00
Supplier#000003075	15.00
Supplier#000003335	15.00
Supplier#000005649	15.00
Supplier#000006027	15.00
Supplier#000006795	15.00
Supplier#000006800	15.00
Supplier#000006824	15.00
Supplier#000007131	15.00
Supplier#000007382	15.00
Supplier#000008913	15.00
Supplier#000009787	15.00
Supplier#000000633	14.00
Supplier#000001960	14.00
Supplier#000002323	14.00
Supplier#000002490	14.00
Supplier#000002993	14.00
Supplier#000003101	14.00
Supplier#000004489	14.00
Supplier#000005435	14.00
Supplier#000005583	14.00
Supplier#000005774	14.00
Supplier#000007579	14.00
Supplier#000008180	14.00
Supplier#000008695	14.00
Supplier#000009224	14.00
Supplier#000000357	13.00
Supplier#000000436	13.00
Supplier#000000610	13.00
Supplier#000000788	13.00
Supplier#000000889	13.00
Supplier#000001062	13.00
Supplier#000001498	13.00
Supplier#000002056	13.00
Supplier#000002312	13.00
Supplier#000002344	13.00
Supplier#000002596	13.00
Supplier#000002615	13.00
Supplier#000002978	13.00
Supplier#000003048	13.00
Supplier#000003234	13.00
Supplier#000003727	13.00
Supplier#000003806	13.00
Supplier#000004472	13.00
Supplier#000005236	13.00
Supplier#000005906	13.00
Supplier#000006241	13.00
Supplier#000006326	13.00
Supplier#000006384	13.00
Supplier#000006394	13.00
Supplier#000006624	13.00
Supplier#000006629	13.00
Supplier#000006682	13.00
Supplier#000006737	13.00
Supplier#000006825	13.00
Supplier#000007021	13.00
Supplier#000007417	13.00
Supplier#000007497	13.00
Supplier#000007602	13.00
Supplier#000008134	13.00
Supplier#000008234	13.00
Supplier#000009435	13.00
Supplier#000009436	13.00
Supplier#000009564	13.00
Supplier#000009896	13.00
Supplier#000000379	12.00
Supplier#000000673	12.00
Supplier#000000762	12.00
Supplier#000000811	12.00
Supplier#000000821	12.00
Supplier#000001337	12.00
Supplier#000001916	12.00
Supplier#000001925	12.00
Supplier#000002039	12.00
Supplier#000002357	12.00
Supplier#000002483	12.00

```
100 rows processed.
Statement Processed in 28.00 seconds.

Ended Executing this Query at Sat Apr  7 08:10:14 2001


Query Started at 986656186.16
Query Ended at 986656214.16
Query Processed in 28.00 seconds


SQL statements processed: 1
Queries processed: 1

====================================================
```

qual22.v1

```
====================================================
Begin Execution at Sat Apr  7 08:10:14 2001


-- using default substitutions

select
cntrycode,
count(*) as numcust,
sum(c_acctbal) as totacctbal
from
(
select
substr(c_phone, 1, 2) as cntrycode,
c_acctbal
from
customer
where
substr(c_phone,1, 2) in
('13', '31', '23', '29', '30', '18', '17')
and c_acctbal > (
select
avg(c_acctbal)
from
```

```
customer
where
c_acctbal > 0.00
and substr(c_phone, 1, 2) in
('13', '31', '23', '29', '30', '18', '17')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode

CNTRYCODE NUMCUST               TOTACCTBAL
13        888.00                6737713.99
17        861.00                6460573.72
18        964.00                7236687.40
23        892.00                6701457.95
29        948.00                7158866.63
30        909.00                6808436.13
31        922.00                6806670.18


7 rows processed.
Statement Processed in 4.72 seconds.

Ended Executing this Query at Sat Apr  7 08:10:19 2001


Query Started at 986656214.46
Query Ended at 986656219.19
Query Processed in 4.72 seconds


SQL statements processed: 1
Queries processed: 1
```

Appendix D. Seed and Query Substitution Parameters

This Appendix contains Seed values and substitution parameters for each stream

seed values

```
session 00 324063752
session 01 324063753
session 02 324063754
session 03 324063755
session 04 324063756
session 05 324063757
session 06 324063758
session 07 324063759
session 08 324063760
```

stream 00 substitution parameters

```
q1 69
q2 38                   TIN             MIDDLE EAST
q3 HOUSEHOLD            1995-03-28
q4 1997-06-01
q5 ASIA
q6 0.03
q7 FRANCE               UNITED STATES
q8a UNITED STATES       AMERICA         ECONOMY BRUSHED
STEEL
q9 seashell
q10 1993-05-01
q11 INDONESIA
q12 FOB                 REG AIR
q13 express             packages
q14 1993-04-01
q15 1993-07-01
q16 Brand#31            MEDIUM ANODIZED (14, 29, 46,
36, 23, 3, 18, 6)
q17 Brand#12            LG JAR
q18 313
q19 Brand#22            8               Brand#54
      16
q20 firebrick           1997-01-01      CHINA
q21 UNITED KINGDOM
q22 ('28', '21', '18', '10', '23', '20', '29')
```

stream 01 substitution parameters

```
q1 77
q2 25                   COPPER                  ASIA
q3 AUTOMOBILE           1995-03-14
q4 1995-03-01
q5 EUROPE
q6 0.09
q7 UNITED KINGDOM       MOZAMBIQUE
q8a MOZAMBIQUE                  AFRICA
       ECONOMY PLATED COPPER
q9 rose
q10 1994-02-01
q11 RUSSIA
q12 MAIL                SHIP
q13 express             packages
q14 1993-07-01
q15 1996-01-01
q16 Brand#21            ECONOMY PLATED (26, 4, 31, 15,
8, 12, 6, 32)
q17 Brand#14            LG PKG
q18 314
q19 Brand#24            3               Brand#42
      17
q20 plum                1996-01-01      GERMANY
q21 MOROCCO
q22 ('33', '20', '11', '12', '23', '31', '34')
```

stream 02 substitution parameters

```
q1 85
q2 13                   STEEL           MIDDLE EAST
q3 FURNITURE            1995-03-30
q4 1997-09-01
q5 MIDDLE EAST
q6 0.06
q7 MOROCCO              INDIA
q8a INDIA               ASIA            ECONOMY ANODIZED
COPPER
q9 pink
q10 1994-11-01
q11 IRAN
q12 TRUCK               SHIP
q13 express             packages
q14 1993-10-01
q15 1993-10-01
q16 Brand#51            STANDARD POLISHED (20, 10, 5,
45, 26, 47, 31, 36)
q17 Brand#11            MED CASE
q18 312
q19 Brand#31            18              Brand#21
      18
q20 burlywood           1994-01-01      RUSSIA
q21 GERMANY
q22 ('19', '12', '33', '10', '13', '24', '21')
```

stream 03 substitution parameters

```
q1 93
q2 1                    BRASS           ASIA
q3 MACHINERY            1995-03-16
q4 1995-06-01
q5 AFRICA
q6 0.04
q7 GERMANY              ALGERIA
q8a ALGERIA             AFRICA
q9 olive
q10 1993-08-01
q11 UNITED KINGDOM
q12 RAIL                SHIP
q13 express             packages
q14 1994-02-01
q15 1996-05-01
q16 Brand#31            LARGE ANODIZED          (37, 27,
49, 33, 13, 44, 14, 41)
q17 Brand#13            MED JAR
q18 314
q19 Brand#33            3               Brand#53
      19
q20 medium              1997-01-01      JAPAN
q21 UNITED STATES
q22 ('13', '21', '32', '17', '33', '15', '26')
```

stream 04 substitution parameters

```
q1 101
q2 39                   NICKEL                  AFRICA
q3 FURNITURE            1995-03-01
q4 1993-03-01
q5 AMERICA
q6 0.09
q7 UNITED STATES        PERU
q8a PERU                AMERICA                 LARGE
BURNISHED COPPER
q9 midnight
q10 1994-05-01
q11 IRAQ
q12 AIR                 SHIP
q13 express             requests
q14 1994-05-01
q15 1994-01-01
q16 Brand#21            PROMO BURNISHED         (27, 48,
31, 9, 10, 37, 32, 21)
q17 Brand#15            MED PKG
q18 315
q19 Brand#35            9               Brand#41
      20
q20 violet              1996-01-01      BRAZIL
q21 PERU
q22 ('12', '16', '28', '10', '34', '33', '18')
```

stream 05 substitution parameters

```
q1 109
q2 26                   COPPER                  ASIA
q3 MACHINERY            1995-03-18
q4 1995-10-01
q5 ASIA
q6 0.07
q7 MOZAMBIQUE           INDONESIA
```

```
q8a INDONESIA          ASIA           MEDIUM BRUSHED
COPPER
q9 lime
q10 1993-03-01
q11 UNITED STATES
q12 SHIP               REG AIR
q13 special            requests
q14 1994-08-01
q15 1996-08-01
q16 Brand#51           SMALL POLISHED        (7, 18,
1, 33, 23, 29, 17, 30)
q17 Brand#12           JUMBO CASE
q18 313
q19 Brand#42           4              Brand#24
       10
q20 grey               1994-01-01     MOZAMBIQUE
q21 INDONESIA
q22 ('14', '13', '21', '12', '29', '15', '31')
=================================================
```

stream 06 substitution parameters

```
=================================================
q1 117
q2 14                  STEEL          AFRICA
q3 BUILDING            1995-03-03
q4 1993-07-01
q5 EUROPE
q6 0.04
q7 INDIA               ARGENTINA
q8a ARGENTINA          AMERICA               MEDIUM
POLISHED TIN
q9 khaki
q10 1993-12-01
q11 JAPAN
q12 FOB                REG AIR
q13 special            requests
q14 1994-11-01
q15 1994-05-01
q16 Brand#31           LARGE BRUSHED  (34, 39, 2, 4,
32, 36, 43, 30)
q17 Brand#14           JUMBO JAR
q18 314
q19 Brand#44           9              Brand#12
       11
q20 saddle             1993-01-01     FRANCE
q21 ARGENTINA
q22 ('16', '11', '19', '31', '24', '23', '32')
=================================================
```

stream 06 substitution parameters

```
=================================================
q1 117
q2 14                  STEEL          AFRICA
q3 BUILDING            1995-03-03
q4 1993-07-01
q5 EUROPE
q6 0.04
q7 INDIA               ARGENTINA
q8a ARGENTINA          AMERICA               MEDIUM
POLISHED TIN
q9 khaki
q10 1993-12-01
q11 JAPAN
q12 FOB                REG AIR
q13 special            requests
q14 1994-11-01
q15 1994-05-01
q16 Brand#31           LARGE BRUSHED  (34, 39, 2, 4,
32, 36, 43, 30)
q17 Brand#14           JUMBO JAR
q18 314
q19 Brand#44           9              Brand#12
       11
q20 saddle             1993-01-01     FRANCE
q21 ARGENTINA
q22 ('16', '11', '19', '31', '24', '23', '32')
=================================================
```

stream 07 substitution parameters

```
=================================================
q1 64
q2 2                   BRASS          EUROPE
q3 MACHINERY           1995-03-20
q4 1996-02-01
q5 MIDDLE EAST
q6 0.02
q7 ALGERIA             CHINA
q8a CHINA              ASIA           MEDIUM BURNISHED
TIN
q9 green
q10 1994-09-01
q11 ALGERIA
q12 MAIL               REG AIR
q13 special            requests
q14 1995-02-01
q15 1996-12-01
q16 Brand#21           STANDARD BURNISHED    (8, 10,
5, 27, 49, 13, 39, 18)
q17 Brand#11           JUMBO CAN
q18 312
q19 Brand#42           4              Brand#45
       12
q20 cream              1996-01-01     VIETNAM
q21 CHINA
q22 ('15', '11', '16', '29', '14', '12', '18')
=================================================
```

stream 08 substitution parameters

```
=================================================
q1 72
q2 40                  NICKEL                AFRICA
q3 BUILDING            1995-03-05
q4 1993-11-01
q5 AMERICA
q6 0.07
q7 MOZAMBIQUE          IRAN
q8a IRAN               MIDDLE EAST    SMALL BRUSHED
TIN
q9 floral
q10 1993-06-01
q11 JORDAN
q12 TRUCK              REG AIR
q13 special            requests
q14 1995-06-01
q15 1994-08-01
q16 Brand#51           MEDIUM PLATED  (23, 35, 39, 48,
46, 6, 19, 14)
q17 Brand#13           WRAP CASE
q18 313
q19 Brand#54           10             Brand#54
       10
q20 olive              1995-01-01     IRAN
q21 IRAN
q22 ('24', '12', '13', '21', '19', '26', '28')
```

Appendix E. Implementation−Specific Layer/Driver Code

```
=====================================================
```

audit.sh

```
=====================================================
#!/bin/ksh
. $KIT_DIR/env
#
# Get Params
set −x
FIRST="yes"
SECOND="yes"

if [ "$1" = "first" ] ; then
   SECOND="no"
else
  if [ "$1" = "second" ] ; then
     FIRST="no"
  fi
fi

if [ "$FIRST" = "yes" ]; then
  # Bounce instance on first node  before first run
  ./tshut
  sleep 60
  ./tstart

 # Bounce instance on second node  before first run
  rsh −n rep2 /export/home/oracle/tshut_rep2
  sleep 60
  rsh −n rep2 /export/home/oracle/tstart_rep2


  echo "Starting first run"
  date

  # first run
  ./audit_stream.sh 3000 first

  echo " Done with first run"
  date

  echo
  echo
fi

if [ "$SECOND" = "yes" ]; then
  # Bounce instance on first node  before second run
  ./tshut
  sleep 60
  ./tstart

# Bounce instance on first node  before second run
  rsh −n rep2 /export/home/oracle/tshut_rep2
  sleep 60
  rsh −n rep2 /export/home/oracle/tstart_rep2


  echo "Starting second run"
  date

  # second run
  ./audit_stream.sh 3000 second

  echo " Done with second run"
  date
fi

wait
./tshut
rsh −n rep2 /export/home/oracle/tshut_rep2

=====================================================
```

audit_stream.sh

```
=====================================================
#!/bin/ksh
. $KIT_DIR/env
#ECHO=/bin/echo
SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of
the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${QGEN_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

RUN_ID_FILE=${KIT_DIR}/r_id

UPD_SQL=${UPD_DIR}/sql
UPD_SPT=${UPD_DIR}/scripts
UPD_SRC=${UPD_DIR}/source
UPD_DAT=${UPD_DIR}/data

TPCD_BIN=${KIT_DIR}/audit/bin
TPCD_LOG=${KIT_DIR}/audit/log
TPCD_RPT=${KIT_DIR}/audit/rpt

OUT=${KIT_DIR}/audit/out
GTIME=${SRC_DIR}/gtime
SEED_FILE=${KIT_DIR}/audit/seed

DF=/dev/null
HID=1
INTERVAL=60
COUNT=1200

# The defaults

USER="tpcd/tpcd"
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 $U2PROG."
echo "−o               : Collect Oracle statistics."
echo "−s               : Collect System statistics."
echo "−u <user/passwd> : User/Password. Default is
tpcd/tpcd."
echo "−h               : Displays this message."
}
set −− ‘getopt "p:u1:u2:osu:h" "$@"‘ || 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 "2" ]
then
  usage
  exit 1
fi

SF=$1
PARA=$2

if [ $PARA = first ]
then
  UF_SET=1
  START_SET=1
  STOP_SET=8
  START_SET_UPDATE=2
```

```
  STOP_SET_UPDATE=9
else
  UF_SET=12
  START_SET=1
  STOP_SET=8
  START_SET_UPDATE=13
  STOP_SET_UPDATE=20
fi

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

echo "TPC-H Power Test Run `date`" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "RUNID is $RUN_ID" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo ""  >>  ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

TPCD_LOG_FILE=${TPCD_LOG}/query.${RUN_ID}.log
TPCD_RPT_FILE=${TPCD_RPT}/query.${RUN_ID}.rpt
QRY_FILE=${QRY_DIR}/qtemp.${RUN_ID}.${SF}

echo "Generates query template file with given seed
for stream 0"
echo "Generates query template file with given seed
for stream 0" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

${QGEN} -c -r `cat $SEED_FILE` -p 0 -s ${SF} >
${QRY_FILE}

echo "Done generating query template file with given
seed for stream 0"
echo "Done generating query template file with given
seed for stream 0" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

START=`$GTIME`
echo "TPC-H Power Test Execution starts at $START"
echo "TPC-H Power Test Execution starts at $START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo ""
echo "" >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

# Execute UF1

SDATE=`date`
echo "Start UF1 at ${SDATE}"
echo "Start UF1\t${SDATE}"
echo "Start UF1\t${SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "Start UF1\t${SDATE}" >>
${OUT}/uf1.${RUN_ID}.${HID} 2>&1

UF1_START=`$GTIME`
echo "Start time for UF1 is $UF1_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

${ECHO} ${UPD_SPT}/runuf1.sh -u ${USER} ${RUN_ID}
${SF} ${UF_SET} >> ${OUT}/uf1.${RUN_ID}.${HID} 2>&1

# Execute Query Stream

UF1_END=`${GTIME}`
echo "End time for UF1 is $UF1_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

EDATE=`date`
echo "End UF1. Start Query Stream at ${EDATE}"
echo "End UF1. Start Query Stream at ${EDATE}" >>
${OUT}/uf1.${RUN_ID}.${HID} 2>&1
echo "" >>  ${OUT}/uf1.${RUN_ID}.${HID} 2>&1
echo "Start Query\t${EDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

${QPROG} tpcd/tpcd q${QRY_FILE} l${TPCD_LOG_FILE}
r${TPCD_RPT_FILE}  >> ${OUT}/qs.${RUN_ID}.${HID} 2>&1

# Execute UF2

SDATE=`date`
echo "End Query Stream. Start UF2 at ${SDATE}"
echo "End Query Stream. Start UF2 at ${SDATE}" >>
${OUT}/uf2.${RUN_ID}.${HID} 2>&1

echo "Start UF2\t${SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
UF2_START=`${GTIME}`
echo "Start time for UF2 is $UF2_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

${ECHO} ${UPD_SPT}/runuf2.sh -u ${USER} ${RUN_ID}
${SF} ${UF_SET} >> ${OUT}/uf2.${RUN_ID}.${HID} 2>&1

UF2_END=`${GTIME}`
echo "End time for UF2 is $UF2_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
EDATE=`date`
echo "END UF2 ${EDATE}"
echo "End UF2\t${EDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "End UF2\t${EDATE}" >>
${OUT}/uf2.${RUN_ID}.${HID} 2>&1

END=`${GTIME}`

echo "TPC-D Power Test Execution ends at $END"
MEA_INT=`echo $END - $START | bc`
echo "Measurement Interval is $MEA_INT"

UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo "Elapsed Time for Update Function is $UF1_TIME"
echo "-- UF1" >> ${TPCD_RPT_FILE}
echo "$UF1_TIME" >> ${TPCD_RPT_FILE}

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
echo "Elapsed Time for Update Function is $UF2_TIME"
echo "-- UF2" >> ${TPCD_RPT_FILE}
echo "$UF2_TIME" >> ${TPCD_RPT_FILE}

echo ""
echo "-----------------------------------------------"
echo ""

echo "Power Test completed at `date`" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt


${SRC_DIR}/metric ${SF} < ${TPCD_RPT_FILE} \
        >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

THRUPUT=`expr 22 \* 3600 \* ${SF}`
THRUPUT=`echo "scale=2\n${THRUPUT}/${MEA_INT}" | bc`
echo "Real Throughput Metric is: $THRUPUT"
echo "Real Throughput Metric is: $THRUPUT" >> \
         ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt



i=$START_SET
PSEED=`cat $SEED_FILE`

while [ $i -le $STOP_SET ]; do
  TPCD_LOG_FILE=${TPCD_LOG}/query.${RUN_ID}_${i}.log
  TPCD_RPT_FILE=${TPCD_RPT}/query.${RUN_ID}_${i}.rpt
  QRY_FILE=${QRY_DIR}/qtemp.${RUN_ID}_${i}.${SF}

  echo "Generate query files for STREAM ${i}"

  PSEED=`expr $PSEED + 1`
  ${QGEN} -c -r ${PSEED} -p ${i} -s ${SF} >
${QRY_FILE}

  i=`expr $i + 1`
done

echo "Done creating multiple streams"

echo

M_START=`${GTIME}`
echo "Multiple-stream starts at $M_START"
echo "Multiple-stream starts at $M_START"  >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

rm -f /tmp/th_pipe1
mknod /tmp/th_pipe1 p
rm -f /tmp/th_pipe2
mknod /tmp/th_pipe2 p

i=$START_SET

while [ $i -le $STOP_SET ]; do
  echo "Start Query\t${M_SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}_${i}.${HID}.rpt
```

```
  TPCD_LOG_FILE=${TPCD_LOG}/query.${RUN_ID}_${i}.log
  TPCD_RPT_FILE=${TPCD_RPT}/query.${RUN_ID}_${i}.rpt
  QRY_FILE=${QRY_DIR}/qtemp.${RUN_ID}_${i}.${SF}
  ${QPROG} tpcd/tpcd q${QRY_FILE} l${TPCD_LOG_FILE} >>
${OUT}/qs.${RUN_ID}_${i}.${HID} 2>&1 &
  i=‘expr $i + 1‘
done
sleep.sh &
(${KIT_DIR}/audit/update_stream.sh ${RUN_ID}
$START_SET_UPDATE $STOP_SET_UPDATE ${SF} >>
${OUT}/us.${RUN_ID}_1.${HID} 2>&1 &)

wait

print > /tmp/th_pipe1
read < /tmp/th_pipe2

M_EDATE=‘date‘
M_END=‘$GTIME‘
echo "Multi−stream ends at $M_END"
echo "Multi−stream end at $M_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

=====================================================
```

runuf1.sh

```
=====================================================
#!/bin/ksh
#
# $Header: runuf1.sh 15−aug−99.18:26:44 mpoess Exp $
#
# runuf1.sh
#
#  Copyright (c) Oracle Corporation 1999. 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      08/15/99 − Creation
#    mpoess      08/15/99 − Creation
#
#. $KIT_DIR/env
#O=${ORACLE_HOME}
TPCD_DIR=/export/home/oracle/kit
UPDATE_DIR=${TPCD_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
LOG_DIR=${UPDATE_DIR}/log
GTIME=${SCRIPT_DIR}/gtime

usage() {
echo ""
echo "runuf1.sh −l [<path name for reports>] −u
[<uid/passwd>]"
echo "          −h <run_id> <scale factor> <pair
number>"
echo "          <parallelism>"
echo ""
echo "run_id       : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number   : The update pair number that this
update function belongs to."
echo "parallelism  : The parallelism of the updates."
echo ""
echo "−l           : Path name for reports on the
update function.  Debug use for UF1 only"
echo "−u           : Userid/Password for Oracle.
Default is tpcd/tpcd."
echo "−h           : To Display this message."
echo ""
}


LOGPATH=.
PASSWD="tpcd/tpcd"

set −− ‘getopt "l:u:p:h" "$@"‘ || usage

while :
do
   case "$1" in
   −u) shift; PASSWD=$1;;
   −l) shift; LOGPATH=$1;;
   −h) usage; exit 0;;
   −−) shift; break;;
   esac
   shift;
done

if [ $# −lt 3 ]
then
   usage
   exit 1
fi

RUN_ID=$1
SF=$2
SETNUM=$3
LDR_PAR=64
PAR_HINT=84
if [ "$PAR_HINT" = "" ]; then
  PAR_HINT=84
fi


i=1
PID=""

# perform the update function 1

START=‘$GTIME‘

# first create the temp tables

svrmgrl << !
connect $PASSWD;

drop table temp_item;
create table temp_item (
    l_shipdate          date ,
    l_orderkey          number,
    l_discount          number ,
    l_extendedprice     number ,
    l_suppkey           number,
    l_quantity          number ,
    l_returnflag        char(1) ,
    l_partkey           number,
    l_linestatus        char(1) ,
    l_tax               number,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipmode          varchar(10) ,
    l_linenumber        number,
    l_shipinstruct      varchar(25) ,
    l_comment           varchar(44)
)
pctfree 1
pctused 99
parallel (degree $PAR_HINT instances 1)
nologging;

drop table temp_ord;
create table temp_ord (
    o_orderdate         date ,
    o_orderkey          number,
    o_custkey           number,
    o_orderpriority     varchar(15) ,
    o_shippriority      number ,
    o_clerk             varchar(15) ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_comment           varchar(79)
)
pctfree 1
pctused 99
parallel (degree $PAR_HINT instances 1)
nologging;

exit;
!
i=1
while [ $i −le $LDR_PAR ]
do

# Kick off sqlldr to load the data into the temporary
staging tables.
```

```
   sqlldr userid=$PASSWD
control=${SCRIPT_DIR}/tempitem.ctl \
   log=${LOG_DIR}/ti${i}.log \

data=${UPDATE_DIR}/data/lineitem.tbl.u${SETNUM}.${i} \
   direct=true parallel=true &

   sqlldr userid=$PASSWD
control=${SCRIPT_DIR}/tempord.ctl \
   log=${LOG_DIR}/to${i}.log \
   data=${UPDATE_DIR}/data/orders.tbl.u${SETNUM}.${i}
\
   direct=true parallel=true &

   i=‘expr $i + 1‘

done
# All sqlldr processes have started, now wait for all
them to finish.

wait

END_LOAD=‘$GTIME‘
# now do the insert and then drop the staging tables
svrmgrl << !
connect $PASSWD;

set timing on;
alter session force parallel dml parallel (degree
$PAR_HINT);
alter session set isolation_level=serializable;
alter session set parallel_instance_group = ’groupb’;
analyze table temp_ord estimate statistics sample 100
rows;
analyze table temp_item estimate statistics sample 100
rows;

commit;
insert /*+ PARALLEL(orders,$PAR_HINT) */
into orders (select * from temp_ord);

insert /*+ PARALLEL(lineitem,$PAR_HINT) */
into lineitem (select * from temp_item);

commit;
exit;
!


svrmgrl << !
connect $PASSWD;
drop table temp_item;
drop table temp_ord;
exit;

!

END=‘$GTIME‘

# Done

echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Load  Time is ‘echo $END_LOAD − $START | bc‘"
echo "Elapsed Time is ‘echo $END − $START | bc‘"
echo ""
```

==========

runuf2.sh

==========

```
#!/bin/ksh
#
# $Header: runuf2.sh 15−aug−99.18:27:21 mpoess Exp $
#
# runuf2.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights
Reserved.
#
#    NAME
#      runuf2.sh − <one−line expansion of the name>
#
# DESCRIPTION
#    runuf2.sh [−u <uid/passwd to login>] [−p
<program>] <run_id>
#              <scale factor> <pair number>
<parallelism>
# USAGE
#    To execute UF2.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/15/99 − Creation
#    mpoess      08/15/99 − Creation
#
#. $KIT_DIR/env
#O=${ORACLE_HOME}
TPCD_DIR=/export/home/oracle/kit
UPDATE_DIR=${TPCD_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
GTIME=${SCRIPT_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log

usage() {
echo ""
echo "runuf2.sh [−u <user/passwd>] <run_id> <scale
factor> <pair number> <parallelism>"
echo ""
echo "run_id       : Run_ID of this update run."
echo "scale factor : Scale Factor of the database."
echo "set number   : The update pair number that this
update function belongs to."
echo "parallelism  : The parallelism of the updates."
echo ""
echo "−u           : Userid/Password for Oracle.
Default is tpcd/tpcd"
echo "−h           : To Display this message."
echo ""
}

PASSWD="tpcd/tpcd"

PAR_HINT=84
if [ "$PAR_HINT" = "" ]; then
  PAR_HINT=84
fi

set −− ‘getopt "u:p:h" "$@"‘ || usage

while :
do
   case "$1" in
   −u) shift; PASSWD=$1;;
   −h) usage; exit 0;;
   −−) shift; break;;
   esac
   shift;
done

if [ $# −lt 3 ]
then
   usage
   exit 1
fi

RUN_ID=$1
SF=$2
SETNUM=$3
PAR=64

i=1
PID=""

START=‘$GTIME‘

# first create the temp tables

svrmgrl << !
connect $PASSWD;

drop table temp_okey;
create table temp_okey (
    t_orderkey           number
)
pctfree 1
pctused 99
parallel (degree $PAR_HINT instances 1)
nologging;

exit;
!
# All processes have started, now wait for all the
update processes to finish.

wait
while [ $i −le $PAR ]
do
```

```
  sqlldr userid=$PASSWD
control=${SCRIPT_DIR}/tempokey.ctl \
  log=${LOG_DIR}/ti${i}.log \
  data=${UPDATE_DIR}/data/delete.${SETNUM}.${i} \
  direct=true parallel=true &

  i=‘expr $i + 1‘

done
wait
# now build the unique index on orders temp table.
svrmgrl << !
connect $PASSWD;
alter session set isolation_level = serializable;
set timing on;
drop index t_iorderkey;
create unique index t_iorderkey
on temp_okey (t_orderkey)
pctfree 2
initrans 10
compute statistics
storage (freelists 99 freelist groups 2)
parallel ( degree $PAR_HINT instances 1);
alter index t_iorderkey parallel (degree $PAR_HINT
instances 1);
exit;
!

svrmgrl << !
connect $PASSWD;

set timing on;
alter session force parallel dml parallel (degree
$PAR_HINT);
alter session set isolation_level=serializable;
alter session set parallel_instance_group = ’groupb’;
analyze table temp_okey estimate statistics sample 100
rows;

commit;
delete from (select /*+ ordered index(o) use_nl(o)
parallel(o, $PAR_HINT,1) index(orders, o_orderkey)*/
o.rowid from orders o,    temp_okey t  where
o.o_orderkey = t.t_orderkey order by 1);
delete from (select /*+ ordered index(l) use_nl(l)
parallel(l, $PAR_HINT,1) index(lineitem, l_orderkey)*/
l.rowid from lineitem l,temp_okey t where l.l_orderkey
= t.t_orderkey order by 1);

commit;
exit;
!

svrmgrl << !
connect $PASSWD;

drop table temp_okey;
exit;

!
#
END=‘$GTIME‘

# Done

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

split_li.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: split_li.c 15−aug−99.18:28:32 mpoess Exp
$ ";
#endif /* RCSID */

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

/*

   NAME
     split_li.c − <one−line expansion of the name>

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

   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>

   RETURNS
     <function return values, for .c file with single
function>

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   mpoess      08/15/99 − Creation
   mpoess      08/15/99 − Creation

*/

#include <stdio.h>

extern int     errno;
extern char     *optarg;
#define OPTARG "i:o:r:s:"

int     getline(FILE *ifp, char *buf)
{
int     idx = 0;
int     rc;

        do {
                rc = fscanf(ifp, "%c", &buf[idx]);
        } while ((buf[idx++] != ’\n’) && (rc != EOF));
        buf[idx] = 0;
        return (rc);
}

main(int argc, char **argv)
{
char    c;
char    *ifile;
char    *ofile;
char    *rfile;
int     splits;
int     bkey;
int     ckey;
int     ekey;
int     i, sav_buf;
char    buf[1024];
char    c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11;
char    ofile_buf[1024];
FILE    *ifp, *ofp, *rfp;
int     done = 0;
int     rc;
int     *range_array;

        while ((c = getopt(argc, argv, OPTARG)) !=
(char)−1) {
                switch (c) {
                        case ’i’:
                                ifile = optarg;
                                break;
                        case ’o’:
                                ofile = optarg;
                                break;
                        case ’r’:
                                rfile = optarg;
                                break;
                        case ’s’:
                                splits = atoi(optarg);
                                break;
                        default:
                                fprintf (stderr,
"usage: −i ifile −o ofile −r range_file −s
number_of_splits\n");
                                exit(0);
                }
        }
        if ((ifp=fopen(ifile, "r")) == (FILE *)NULL) {
                fprintf (stderr, "fopen %s failed,
%d\n", ifile, errno);
                exit (0);
        }
        if ((rfp=fopen(rfile, "r")) == (FILE *)NULL) {
                fprintf (stderr, "fopen %s failed,
```

```
%d\n", rfile, errno);
                exit (0);
        }
        range_array = (int *)malloc(sizeof(int) *
splits * 2);
        for (i = 0; i < splits; i++) {
                if ((rc = getline(rfp, buf)) != EOF)
                        sscanf(buf, "%d|%d",
&range_array[2*i], &range_array[(2*i)+1]);
                else fprintf (stderr, "missing key
ranges in %s\n", rfile);
        }
        fclose(rfp);
        sav_buf = 0;
        for (i = 0; i < splits; i++) {
                sprintf (ofile_buf, "%s.%d", ofile,
i+1);
                if ((ofp=fopen(ofile_buf, "w")) ==
(FILE *)NULL) {
                        fprintf (stderr, "fopen %s
failed, %d\n",  ofile_buf, errno);
                        exit (0);
                }
                done = 0;
                bkey = range_array[2*i];
                ekey = range_array[(2*i)+1];
                while (!done) {
                        rc = 0;
                        if (!sav_buf)
                                rc = getline(ifp, buf);
                        sav_buf = 0;
                        if (rc != EOF) {
                                sscanf(buf,
"%c%c%c%c%c%c%c%c%c%c%d", &c1, &c2, &c3, &c4, &c5,
&c6, &c7, &c8, &c9, &c10, &c11, &ckey);
                                if ((ckey >= bkey) &&
(ckey <= ekey))
                                        fprintf(ofp,
"%s", buf);
                                else if (ckey > ekey) {
                                        sav_buf = 1;
                                        done = 1;
                                }
                        } else done = 1;
                }
                fclose(ofp);
        }
        fclose(ifp);
}
```

==========

split_li.sh

==========

```
#!/bin/ksh
#
# $Header: split_li.sh 15-aug-99.18:27:49 mpoess Exp $
#
# split_li.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights
Reserved.
#
#    NAME
#      split_li.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      08/15/99 - Creation
#    mpoess      08/15/99 - Creation
#

. $KIT_DIR/env

# $1 LI flatfile
# $2 ORDERS flatfile base
# $3 number of splits
BIN_DIR=`pwd`

i=1
rm -f ${2}.range
while [ $i -le $3 ]
do
  BKEY=`head -1 ${2}.${i} | cut -f2 -d\|`
  EKEY=`tail -1 ${2}.${i} | cut -f2 -d\|`
  print -n $BKEY >> ${2}.range
  print -n '|' >> ${2}.range
  print $EKEY >> ${2}.range
  i=`expr $i + 1`
done
$BIN_DIR/split_li -i $1 -o $1 -r ${2}.range -s $3
rm -f ${2}.range
```

==========

split_li_mp.c

==========

```
#ifdef RCSID
static char *RCSid =
   "$Header: split_li.c 15-aug-99.18:28:32 mpoess Exp
$ ";
#endif /* RCSID */

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

/*

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

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

   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>

   RETURNS
     <function return values, for .c file with single
function>

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   mpoess      08/15/99 - Creation
   mpoess      08/15/99 - Creation

*/

#include <stdio.h>

extern int     errno;
extern char     *optarg;
#define OPTARG "i:o:r:s:"

int     getline(FILE *ifp, char *buf)
{
int     idx = 0;
int     rc;

        do {
                rc = fscanf(ifp, "%c", &buf[idx]);
        } while ((buf[idx++] != '\n') && (rc != EOF));
        buf[idx] = 0;
        return (rc);
}

main(int argc, char **argv)
{
char    c;
char    *ifile;
char    *ofile;
char    *rfile;
int     splits;
int     bkey;
int     ckey;
int     ekey;
int     i, sav_buf;
char    buf[1024];
char    c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11;
char    ofile_buf[1024];
FILE    *ifp, *ofp, *rfp;
int     done = 0;
int     rc;
int     range_array[1000];
```

```
        while ((c = getopt(argc, argv, OPTARG)) !=
(char)−1) {
                switch (c) {
                        case ’i’:
                                ifile = optarg;
                                break;
                        case ’o’:
                                ofile = optarg;
                                break;
                        case ’r’:
                                rfile = optarg;
                                break;
                        case ’s’:
                                splits = atoi(optarg);
                                break;
                        default:
                                fprintf (stderr,
"usage: −i ifile −o ofile −r range_file −s
number_of_splits\n");
                                exit(0);
                }
        }
        if ((ifp=fopen(ifile, "r")) == (FILE *)NULL) {
                fprintf (stderr, "fopen %s failed,
%d\n", ifile, errno);
                exit (0);
        }
        if ((rfp=fopen(rfile, "r")) == (FILE *)NULL) {
                fprintf (stderr, "fopen %s failed,
%d\n", rfile, errno);
                exit (0);
        }
        /*range_array = (int *)malloc(sizeof(int) *
splits * 2);*/
        for (i = 0; i < splits; i++) {
                if ((rc = getline(rfp, buf)) != EOF)
                        sscanf(buf, "%d|%d",
&range_array[2*i], &range_array[(2*i)+1]);
                else fprintf (stderr, "missing key
ranges in %s\n", rfile);
        }
        fclose(rfp);
        sav_buf = 0;
        for (i = 0; i < splits; i++) {
                sprintf (ofile_buf, "%s.%d", ofile,
i+1);
                if ((ofp=fopen(ofile_buf, "w")) ==
(FILE *)NULL) {
                        fprintf (stderr, "fopen %s
failed, %d\n",  ofile_buf, errno);
                        exit (0);
                }
                done = 0;
                bkey = range_array[2*i];
                ekey = range_array[(2*i)+1];
                while (!done) {
                        rc = 0;
                        if (!sav_buf)
                                rc = getline(ifp, buf);
                        sav_buf = 0;
                        if (rc != EOF) {
                                sscanf(buf,
"%c%c%c%c%c%c%c%c%c%c%d", &c1, &c2, &c3, &c4, &c5,
&c6, &c7, &c8, &c9, &c10, &c11, &ckey);
                                if ((ckey >= bkey) &&
(ckey <= ekey))
                                        fprintf(ofp,
"%s", buf);
                                else if (ckey > ekey) {
                                        sav_buf = 1;
                                        done = 1;
                                }
                        } else done = 1;
                }
                fclose(ofp);
        }
        fclose(ifp);
}
```

splitfiles.c

```
#ifdef RCSID
static char *RCSid =
   "$Header: splitfiles.c 15−aug−99.18:31:25 mpoess
Exp $ ";
#endif /* RCSID */

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

/*

   NAME
     splitfiles.c − <one−line expansion of the name>

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

   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>

   RETURNS
     <function return values, for .c file with single
function>

   NOTES
     <other useful comments, qualifications, etc.>

   MODIFIED   (MM/DD/YY)
   mpoess      08/15/99 − Creation
   mpoess      08/15/99 − Creation

*/

#include <stdio.h>

extern int     errno;
extern char     *optarg;
#define OPTARG "i:n:o:l:"

int     getline(FILE *ifp, char *buf)
{
int     idx = 0;
int     rc;

        do {
                rc = fscanf(ifp, "%c", &buf[idx]);
        } while ((buf[idx++] != ’\n’) && (rc != EOF));
        buf[idx] = 0;
        return (rc);
}

main(int argc, char **argv)
{
char    c;
char    *ifile;
char    *ofile;
char    ofile_buf[1024];
char    buf[1024];
FILE    *ifp, *ofp;
int     i;
int     num_splits;
int     line_ct;
int     rem, l_ct;
int     rc;

        while ((c = getopt(argc, argv, OPTARG)) !=
(char)−1) {
                switch (c) {
                        case ’i’:
                                ifile = optarg;
                                break;
                        case ’o’:
                                ofile = optarg;
                                break;
                        case ’n’:
                                num_splits =
atoi(optarg);
                                break;
                        case ’l’:
                                line_ct =
atoi(optarg);
                                break;

                        default:
                                fprintf (stderr,
"usage: −i ifile −o ofile_base −n number_of_splits −l
line_count_of_ifile\n");
                                exit(0);
                }
        }
        if ((ifp=fopen(ifile, "r")) == (FILE *)NULL) {
```

```
                fprintf (stderr, "fopen %s failed,
%d\n", ifile, errno);
                exit (0);
        }
        rem = line_ct % num_splits;
        for (i = 1; i <= num_splits; i++) {
                sprintf(ofile_buf, "%s.%d", ofile, i);
                if ((ofp=fopen(ofile_buf, "w")) ==
(FILE *)NULL) {
                        fprintf (stderr, "fopen %s
failed, %d\n", ofile_buf, errno);
                        exit (0);
                }
                if (rem) {
                        l_ct = (line_ct / num_splits) +
1;
                        rem--;
                } else l_ct = line_ct / num_splits;
                while ((l_ct) && ((rc = getline(ifp,
buf)) != EOF)) {
                        fprintf(ofp, "%s", buf);
                        l_ct--;
                }
                fclose(ofp);
        }
        fclose(ifp);
}
```

tempitem.ctl

```
unrecoverable
load
append
into table temp_item
append
fields terminated by '|'
(
  l_shipdate            date "yyyy-mm-dd",
  l_orderkey            ,
  l_discount            ,
  l_extendedprice       ,
  l_suppkey             ,
  l_quantity            ,
  l_returnflag          ,
  l_partkey             ,
  l_linestatus          ,
  l_tax                 ,
  l_commitdate          date "yyyy-mm-dd",
  l_receiptdate         date "yyyy-mm-dd",
  l_shipmode            ,
  l_linenumber          ,
  l_shipinstruct        ,
  l_comment
)
```

tempokey.ctl

```
unrecoverable
load
append
into table temp_okey
append
fields terminated by '|'
(
  t_orderkey
)
```

tempord.ctl

```
unrecoverable
load
append
into table temp_ord
append
fields terminated by '|'
(
  o_orderdate           date "yyyy-mm-dd",
  o_orderkey            ,
  o_custkey             ,
  o_orderpriority       ,
  o_shippriority        ,
  o_clerk               ,
  o_orderstatus         ,
  o_totalprice          ,
  o_comment
)
```

update_stream.sh

```
#!/bin/ksh
. $KIT_DIR/env
SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
UPD_SPT=${UPD_DIR}/scripts
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of
the query template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

RUN_ID=$1
START_SET=$2
STOP_SET=$3

TPCD_RPT=${KIT_DIR}/audit/rpt

OUT=${KIT_DIR}/audit/out

GTIME=${SRC_DIR}/gtime
HID=1
SF=$4

START=`$GTIME`
echo "TPC-H Update Stream starts at $START"
echo ""


#waiting for all the query streams to finish first
read < /tmp/th_pipe1

i=$START_SET

while [ $i -le $STOP_SET ]; do

# Execute UF1

SDATE=`date`
echo "Start UF1 ${i} at ${SDATE}"
echo "Start UF1\t${i}\t${SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

UF1_START=`$GTIME`
echo "\t\t\tStart time for UF1 is $UF1_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
${UPD_SPT}/runuf1.sh -u ${USER} ${RUN_ID} ${SF} ${i}
>> ${OUT}/uf1.${RUN_ID}.${HID} 2>&1

UF1_END=`${GTIME}`
echo "\t\t\tEnd time for UF1 is $UF1_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

EDATE=`date`
echo "End UF1\t${i}\t${EDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "End  UF1\t${i}\t${EDATE}" >>
${OUT}/uf1.${RUN_ID}.${HID} 2>&1

# Execute UF2

SDATE=`date`
echo "Start UF2 ${i} at ${SDATE}"
echo "Start UF2\t${i}\t${SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
UF2_START=`${GTIME}`
echo "\t\t\tStart time for UF2 is $UF2_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

${UPD_SPT}/runuf2.sh -u ${USER} ${RUN_ID} ${SF} ${i}
>> ${OUT}/uf2.${RUN_ID}.${HID} 2>&1

UF2_END=`${GTIME}`
echo "\t\t\tEnd time for UF2 is $UF2_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
EDATE=`date`
echo "END UF2 ${i} ${EDATE}"
echo "End UF2\t${i}\t${EDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo " " >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
```

```
END=‘$GTIME‘

i=‘expr $i + 1‘
done

print > /tmp/th_pipe2
```

Appendix F. Activity Between Database Load and Run1

```
When the load finished, the load.sh script
automatically selected a seed
value and saved it.

Then the 2 auditor scripts count.sql and dbtables.sql
were run to validate
that the database structure was correct.

Then the database was shut down and both machines were
rebooted.
```

count.sql

```
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;
```

dbtables.sql

```
set numwidth 25
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;



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;
```

load.sh

```
#!/bin/ksh

SDATE=‘date‘
echo "Setting the random number seed at $SDATE"
PSEED=‘date +%m:%d:%H:%M:%S | sed −e ’s/://g’‘
echo "Using ${PSEED} as seed0"
echo ${PSEED} >
/export/home/oracle/kit/audit/build/seed
cp −p /export/home/oracle/kit/audit/build/seed
/export/home/oracle/kit/audit/seed
echo "Done setting the random number seed at ‘date‘"
>> ${LOG_FILE}

echo running auditors scripts
svrmgrl << !
connect tpcd/tpcd
```

```
spool /export/home/oracle/kit/audit/count.out
@/export/home/oracle/kit/audit/count.sql
spool off
spool /export/home/oracle/kit/audit/dbtables.out
@/export/home/oracle/kit/audit/dbtables.sql
exit
!

echo shutting database down

/export/home/oracle/kit/audit/tshut abort
sleep 60
rsh −n rep2 /export/home/oracle/tshut_rep2 abort

====================================================
```

tshut

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

if [ "$2" != "" −a "$2" != "1" ]; then
  INUM=$2
  if [ −f $ORACLE_HOME/work/t_init$INUM.ora ]; then
    export ORACLE_SID="$ORACLE_SID"$INUM
  fi
fi

if [ "$1" = "abort" ]; then
svrmgrl << !
connect internal
shutdown abort
exit
!
else
svrmgrl << !
connect internal
shutdown immediate
exit
!
fi

====================================================
```

tshut_rep2

```
====================================================
#!/bin/ksh
export ORACLE_HOME=/export/home/oracle/oracle817
export ORACLE_SID=inst2
export
PATH=${PATH}:${ORACLE_HOME}/rdbms/bin:${ORACLE_HOME}/b
in
export
LD_LIBRARY_PATH=${ORACLE_HOME}/rdbms/lib:${ORACLE_HOME
}/lib:"$LD_LIBRARY_PATH"

if [ "$1" = "abort" ]; then
svrmgrl << !
connect internal
shutdown abort
exit
!
else
svrmgrl << !
connect internal
shutdown immediate
exit
!
fi
```

Appendix G. Pricing

The following pages contain the price quotes for the hardware and software included in the FDR.

CAT TECHNOLOGY

131 C Albright Way
Los Gatos, Ca. 95032

SALES QUOTE

Rev. 04/12/01

PO #

Quote # LGOQ2697

Sales Support Rep

Morgan Browne
Phone: (408) 341-0812
Fax: (408) 341-1720
Email: morgan@cattech.com

Account Manager

Morgan Browne
Phone: (408) 341-0812
Fax: (408) 341-1720
Email: morgan@cattech.com

FOB

Terms

Quote To:

Quote Good for 90 Days

Phone:

Ship To:

Sun Microsystems, Inc.
Daryl Madura
8305 S.W. Creekside Place
Beaverton OR, 97008

Item	Mfr.	Part No.	Qty	Description	List Price	Unit Price	Ext. Price
1				HARDWARE			
2	SUN	E10000-4	2	EOL - 13-APR-01 - - Enterprise 10000 Server base cabinet, cage for 16 system boards, peripheral space w/power sequencer, Solaris license for unlimited users, license for SSP software	$0.00	$133,532.61	$267,065.22
3	SUN	C2761A	32	E10000 System Board - MUST CTO/ DROPSHIP	$0.00	$40,801.63	$1,305,652.16
4	SUN	X2580A	128	400-MHZ UltraSPARC module with 8-MB external cache.	$0.00	$11,086.96	$1,419,130.88
5	SUN	X7025A	32	E10000 memory board, no SIMMs	$0.00	$7,391.30	$236,521.60
6	SUN	X7023A	128	1-GB memory expansion (8x128-MB SIMMs)	$0.00	$6,060.87	$775,791.36
7	SUN	X2730A	26	E10000 dual SBus I/O card	$0.00	$5,543.48	$144,130.48
8	SUN	X2731A	6	E10000 dual PCI I/O daughter card	$0.00	$7,418.48	$44,510.88
9	SUN	X2755A	2	Sun Enterprise 10000 System Service Processor: Enterprise 250 Server, 1 400-Mhz processor, 256MB of memory, two 9.1-GB internal disks, 32x CD-ROM, 4-mm DDS tape drive, 3.5 inch, 1.44-MB floppy disk drive & Sun Quad FastEthernet PCI adapter	$0.00	$11,456.52	$22,913.04
10	SUN	X3875A	8	E10000 AC Breaker/Filter assembly	$0.00	$2,217.39	$17,739.12

Item	Mfr.	Part No.	Qty	Description	List Price	Unit Price	Ext. Price
11	SUN	X9685A	16	E10000 48 volt power supply (see Spare part 300-1368)	$0.00	$2,217.39	$35,478.24
12	SUN	X9681A	2	E10000 Power Control Module, includes cables	$0.00	$741.85	$1,483.70
13	SUN	X9671A	32	E10000 fan Tray	$0.00	$1,108.70	$35,478.40
14	SUN	X2722A	2	Sun Enterprise 10000 control board with rack-mounted Ethernet hub	$0.00	$11,086.96	$22,173.92
15	SUN	X3865A	8	Sun Enterprise 10000 power cord, 14 feet (U.S. version)	$0.00	$0.00	$0.00
16	SUN	X6730A	96	FC-100 FC-AL SBus host adapter with one GBIC	$0.00	$1,995.65	$191,582.40
17	SUN	X1074A	12	Cluster SCI PCI adapter	$0.00	$3,326.09	$39,913.08
18	SUN	X3902A	12	CPI-SCI Cable 5m	$0.00	$0.00	$0.00
19	SUN	X1049A	4	Sun Quad FastEthernet with qfe naming	$0.00	$1,474.57	$5,898.28
20	SUN	X1065A	4	SBus Ultra differential Fast/Wide intelligent SCSI-2 host adapter (UDWIS/S)	$0.00	$957.17	$3,828.68
21	SUN	X3800A	2	Power cord for Enterprise expansion cabinet (USA version)	$0.00	$0.00	$0.00
22	SUN	X979A	4	12-meter differential SCSI expansion cable	$0.00	$288.04	$1,152.16
23	SUN	X978A	56	15-meter Fibre Optic Cable	$0.00	$190.22	$10,652.32
24	SUN	X3508A	2	Type-6 country Kit for U.S. (North American)/ Universal/ Canada with Sun Interface	$0.00	$0.00	$0.00
25				SubTotal			$4,581,095.92
26				STORAGE			
27	SUN	SG-ARY154A-218GR5	4	218GB D1000 Installed in Sun StorEdge expansion cabinet, 12x1in.18GB 10k RPM disks, 2 pwr supp, 2 fan trays (4 fans) 4 diff UltraSCSI to host ports, and rack mount rails	$0.00	$12,554.35	$50,217.40
28	SUN	XT3ES-RK-44-1310	1	EOL 20-JUL-01 1310GBT3ES includes, 4xT3 arrays configured in 2 partner groups, 1144GB RAID 5 storage preconfigured as 8 RAID 5 LUNs(8+1), 36x36.4GB 10K RPM FC-AL drives, 4 copper optic adapter, 2x15 fiber optic, 4 unit Interconnect cables, 72	$0.00	$167,856.52	$167,856.52

Item	Mfr.	Part No.	Qty	Description	List Price	Unit Price	Ext. Price
29	SUN	SG-ARY543A-2400G	10	2400GB A5200 Cabinet(22x18.2GB 10k RPM drives) 6-400GB arrays (132drives total) & 4-FC-AL hubs, (4GBICs ea)mntd in 72 cabinet w/2pwr cds, 4-15m fibre optic cbles(factory 10K)	$0.00	$229,500.00	$2,295,000.00
30	SUN	X3858A	20	A3500 Power Cord, domestic	$0.00	$0.00	$0.00
31				SubTotal			$2,513,073.92
32				SERVER SOFTWARE			
33	SUN	SCMMS-210-R99R	2	Component Manager 2.1 Media Kit and Doc. L10N local language version featuring English, French, Japanese, Korean, Simplified Chinese and Traditional Chinese	$0.00	$0.00	$0.00
34	SUN	VVMGS-304-9999	2	Veritas Volume Manager 3.0.4 for A5x00s - Media, Documentation and RTU, please note, Veritas Volume Manager is free with the purchase of Sun StorEdge A5x00 Arrays	$0.00	$0.00	$0.00
35	SUN	SSP9S-330-SAM9	2	E10000 SSP SW 3.3, CD Release	$0.00	$0.00	$0.00
36				SubTotal			$0.00

Abbreviated Terms and Conditions:

* By accepting this quote, Customer agrees to the Terms and Conditions located at http://www.cattech.com/terms/index.html, as amended from time to time.
* This quote is valid for 15 days and subject to revision if/when vendor's prices go up.
* Invoices are generated when product is shipped from third party vendor.
* Customer agrees to pay invoice in full within 30 days of invoice date.
* Customer agrees to pay a late fee of 18% per annum for any past due balance.

Sub Total	$7,094,169.84
Sales Tax	$0.00
Shipping	$0.00
Total	**$7,094,169.84**

Remit To Address:

CAT Technology
Dept. 33237
P.O. Box 39000
San Francisco, CA 94139-3237

Purchase Order Number
(if different from above)

_____

ACCEPTED BY: _____ Date: _____