TPC Benchmark™ D
Full Disclosure Report

BULL SA

Express5800 HV8600 Model 2

With Oracle8*i*

and Windows NT 4.0

First Edition
12-Fev. 1999

Special notices

Express5800 is registered trademarks of Bull company, NEC Corporation, and Packard Bell -NEC.

ORACLE, SQL*Loader, Oracle8i, and Pro*C are registered trademarks of Oracle Corporation.

Windws NT is a trademark of the Microsoft Corporation.

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

PentiumII, Xeon are trademarks or registered trademarks of the Intel Corporation.

Other product names mentioned in this document may be trademarks and/or registered trademarks of their respective companies.

First Edition February 12th, 1999

The information contained in this document has not been submitted to any formal BULL test and is distributed on an "as is " basis without any warranty either express or implied. The use of this information or the implementation of any of this techniques is customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by BULL for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

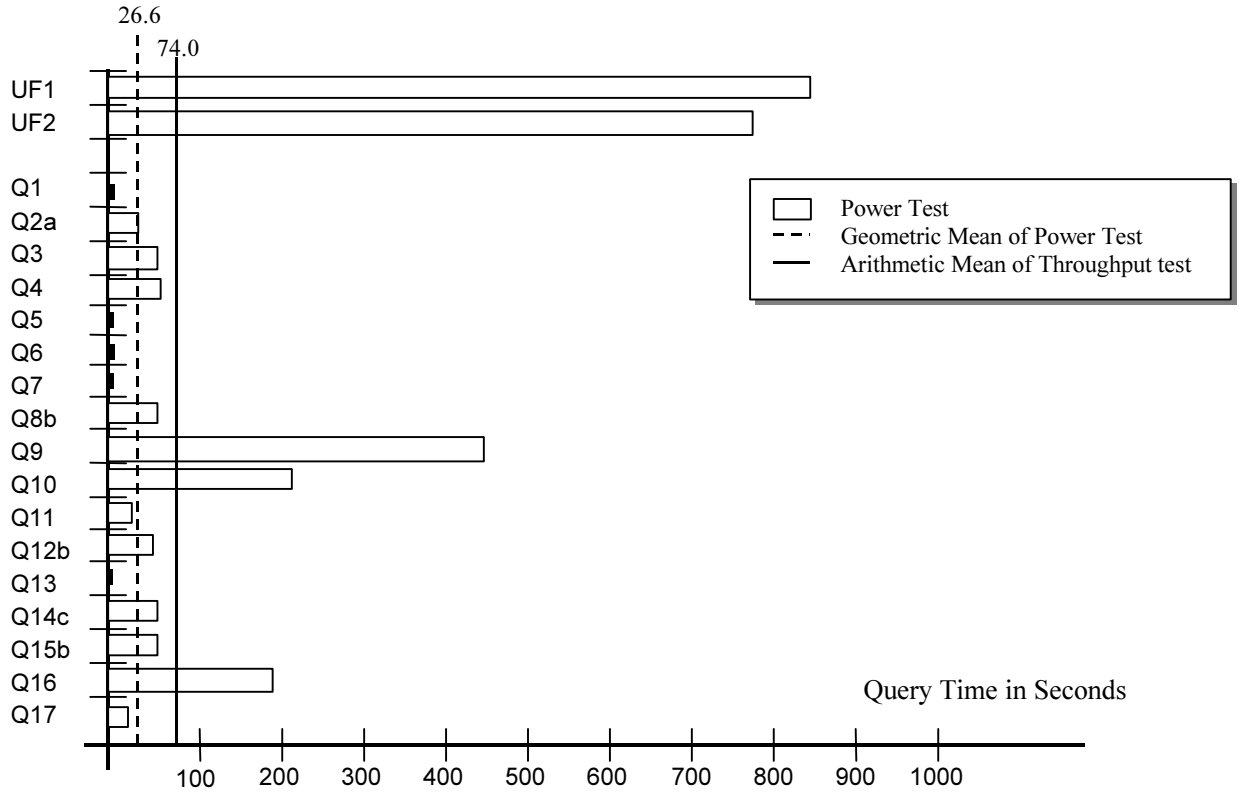
It is possible that this material may contain reference to, or information about, BULL products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that BULL intends to announce such products, programming, or services in your country.

Requests for additional copies of this report should be sent to the following address:

Jean François Lemerre
Bull TPC representative
1 rue de Provence
BP 208
38432 Echirolles Cedex

Bull / ORACLE®	**Express5800 HV8600 Model2 Using Oracle8*i***	TPC-D 1.3.1 Report Date: **12 Feb. 1999**

Total System Cost	TPC-D Power	TPC-D Throughput	Price/Performance	
765,474 Euros	**13,554.3** QppD@100GB	**2,143.66** QthD@100GB	**142.01 Euros** per QphD@100GB	
Database Size	Database Manager	Operation System	Other Software	Availability Date
100GB	**Oracle 8*i* v 8.1.5.1**	**Windows NT Server 4.0 Enterprise Edition**	**MKS Toolkit Visual C++**	**May 18,1999**



Database Load Time = 34 h 16 m	Disk Size/Database Size = 11.6	RAID:N

System Configuration:

- 8 x 450MHz PentiumⅡ Xeon CPUs each with 2MB Cache
- 4Gbytes memory
- 3 x 9.1GB internal drives
- 5 NEC SecuRAID 530 RAID controller with 24 x 9.1GB drives each
- 1 NEC SecuRAID 530 RAID controller with 4 x 9.1GB drives each

Bull / ORACLE	Express5800 HV8600	TPC-D REV 1.3.1 EXECUTIVE SUMMARY				
	Using Oracle8i	Report Date: Feb 12, 99				
Description	**Part Number**	**Pricing**	**Unit Price**	**Quantity**	**Extended Price**	**5-year Maintain. Price**
Server Hardware						
Express5800 HV8600 Dual Pentium Xeon 450MHz/2MB Deskside Model 0 RAID 64MB L3 cache, 32X CD-ROM, 3.5" FDD, no RAM, no HDD, RAID adapter w/ 32MB cache, embedded ETH adapter, Kbd, mouse	HSXH013-002Z	1	36 487	1	36 487	
CPU board with 2 Pentium II Xeon 450MHz/2MB and 64MB L3 cache	CPUH104-200F	1	16 096	1	16 096	
Pentium II Xeon 450MHz/2MB Processor kit (1)	CPUH103-100F	1	5 869	4	23 476	
1GB (4X256MB) 60ns DIMM Memory upgrade kit	CMMH011-000F	1	3 833	4	15 332	
SecuRaid 530, 64 bit PCI Ultra2 SCSI tri-channel RAID card (mode 0,1,5,0+1) model 0, including 0 Mb cache, drivers and documentation	AIF-0690-IN-HU	1	1 832	6	10 992	
32MB cache memory module + battery backup unit for AIF-0690	AMR-0633-IN-00	1	659	6	3 954	
9.1GB LVD HDD (10000rpm) including Hot-Plug carrier	MSUH010-000F	1	1 378	3	4 134	
Z-VISION MR 15"	ZCM-1520-H2-00	1	192	1	192	
Silver Service (4-hour On-Site) for class F servers	GMHH021-S3F4	1		1		2 799
One year Silver Service (4-hour On-Site) extension for class F servers	GMHH021-S1F4	1		2		2 520
				Subtotal	110 663	5 319
Storage Subsystem						
Express5800 ST8000 Tower Model 0	ADE-0002-00-00	1	2 061	17	35 037	
9.1Gb Ultra2 SCSI HDD (5.4ms, 10000rpm, 80 pin, 1") including Hot Plug carrier	ADH-1943-IN-00	1	1 378	124	170 872	
1-meter 68-pin VHD to VHD SCSI-3 data cable	ACN-0015-00-00	1	120	12	1 440	
3-meter 68-pin VHD to VHD SCSI-3 data cable	ACN-0017-00-00	1	151	5	755	
1-channel Ultra SCSI Extender board	AAD-0100-IN-00	1	456	17	7 752	
Single-SCSI bus 1X8 Option Module	AZA-1008-IN-00	1	72	17	1 224	
Silver Service (4-hour On-Site) for ST8000	GMHH021-S3S4	1		17		20 519
One year Silver Service (4-hour On-Site) extension for ST8000	GMHH021-S1S4	1		34		18 462
				Subtotal	217 080	38 981
Server Software						
Windows NT Enterprise Edition (Media & Doc + 25 Client licences) - English	EXSH001-TNEF	1	4 585	1	4 585	
MKS Toolkit for NT	MKS NT	3	508	1	508	
Microsoft Visual C++ Professional Edition	C++	4	563	1	563	
Oracle8i for WinNT	Oracle 8i 8.1.5.1	2	166531	1	166 531	221 244
				Subtotal	172 187	221 244
				TOTAL	499 930	265 544

Notes: All prices in Euros (•)

Pricing: 1-Bull 2-Oracle 3-.Mortice Kern Systems (France) 4- SOS developpers

Audited by Francois Raab of Information Paradigm, Inc

Five-Year Cost of Ownership: 765474
QphD@100GB: 5390,30
Euro/QphD@100GB: 142,01

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

Bull ORACLE®	BULL Express5800 HV8600 Model2 Using Oracle8*i*	TPC-D 1.3.1 Report Date: **12 Feb. 1999**

Numerical Quantity Summary

Measurement Results:

Database Scaling Factor(SF/Size)	= 100GB
Total Data Storage/Database Size	= 11.6
Database Load Time	= 34 hours 16 minutes
Query Stream for Throughput Test	= 0
TPC-D Power Metric (QppD@100GB)	= 13,554.3
TPC-D Throughput Metric (QthD@100GB)	= 2,143.6
Composite QphD@100GB	= 5,390.3
Total System Price Over 5 Years	= 608,305
TPC-D Price/Performance Metric	= 113

Measurement Intervals:

Measurement Interval in Throughput Test (Ts) = 2,855 seconds

Duration of Stream Execution:

Stream ID	Seed	Start Date	Start Time	End Date	End Time
Stream00	454986351	1/29/99	16:55:03	1/29/99	17:42:38
UF1		1/29/99	16:55:03	1/29/99	17:09:23
UF2		1/29/99	17:30:22	1/29/99	17:42:38

TPC-D Timing Intervals (in seconds)

Q1	Q2a	Q3	Q4	Q5	Q6	Q7	Q8b	Q9	Q10
0.2	27.1	42.6	53.8	0.8	0.1	0.6	60.2	456.1	228.1

Q11	Q12b	Q13	Q14c	Q15b	Q16	Q17	UF1	UF2
20.8	42.1	5.8	57.4	56.5	193.4	12.2	859.6	736.7

Information Paradigm

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL

Certified Auditor

Test Sponsors:

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

Jean-François Lemere
Manager, AIX Performance.
BULL S.A.
1, rue de Provence
38230 Echirolles, FRANCE

February 12, 1999

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

Platform: Bull Express 5800 HV8600 Model2
DataBase Manager: Oracle8i Version 8.1.5.1
Operating System: Windows NT Server Enterprise Edition Version 4.0

The results were:

CPU's Speed	Memory	Disks	QppD@100GB	QthD@100GB
Bull Express 5800 HV8600 Model2				
8 x Intel Xeon (450 MHz)	2 MB L2/cpu 4 GB main	124 x 9.1 GB ext. 3 x 9.1 GB int.	13,554.3	2,143.6

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

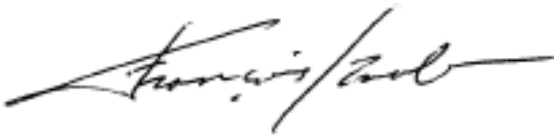
- The TIME table was not used
- The input variables were generated by QGEN
- The database was populated using DBGEN
- The database was maintained by the "Evolve" method
- The throughput metric was computed using the results from the power test
- The ratio between the longest and the shortest query was such that some query times were adjusted
- A compliant implementation specific layer was used
- The query text was produced using compliant variants and minor modifications

- The database records were defined with the proper layout and size
- The database was properly scaled to 100GB and populated accordingly
- The database load time was correctly measured and reported
- The ACID Properties were verified and met
- The reported execution times were correctly measured and reported
- Measurement repeatability was verified
- At least 8 hours 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

TPC Benchmark D Overview

The TPC Benchmark™ D (TPC-D) is a Decision Support benchmark. It is a suite of business-oriented queries and concurrent updates. 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-D evaluates the performance of various Decision Support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-D queries:

- Give answers to real-world business questions
- 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

General Items

Benchmark Sponsor

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

Bull SA and Oracle Corporation are the sponsors of this TPC-D benchmark.

Parameter Settings

Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:

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

This requirement can be satisfied by providing a full list of all parameters and options.

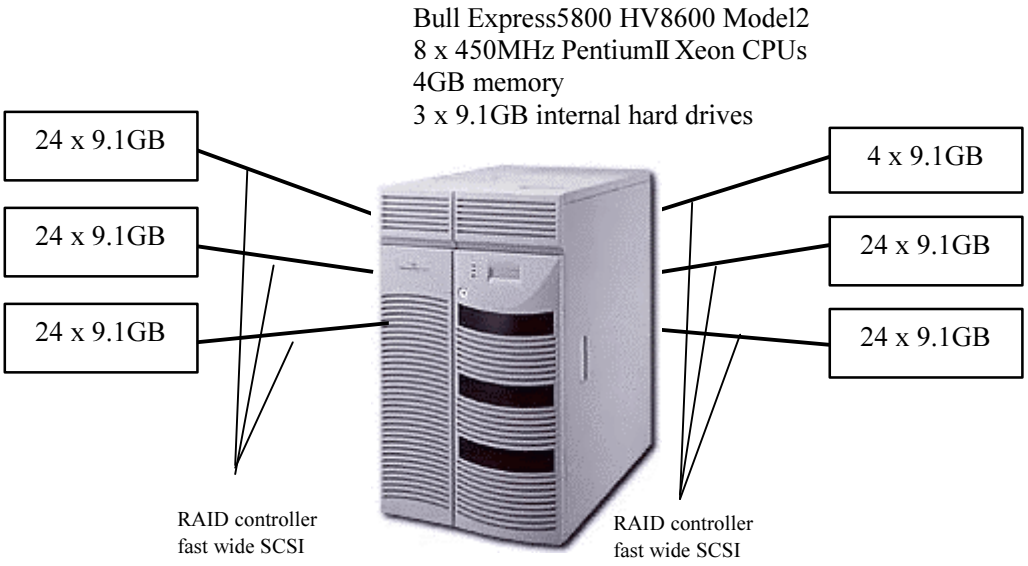
Appendix A contains the WindowsNT and Oracle8i parameters used in this benchmark.

Configuration Diagram

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences. This includes, but is not limited to:

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

The configuration diagram for the tested and priced system is provided on the following page.



Clause 1 Logical Database Design

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-D database.

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 order clustering was used. Column ordering has been modified for some tables. Refer to the table create statements in Appendix B for further details.

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 base tables LINEITEM, ORDERS, PARTS and CUSTOMER, for materialized view logs on LINEITEM and ORDERS, for materialized views MJ_LOCS and MJ_PPSSRN and indexes L_ORED, IJ_L_ODATE, IJ_L_RDATE, IJ_L_PKEY, P_TP, P_CBP and IJ_P_NNPSPS. The details of this partitioning can be understood by examining the syntax of the table, materialized view and index definition statements in Appendix B. Similar partitioning was used in the qualification database, scaled appropriately for its size.

Replication

While there are some restrictions placed upon physical replication of objects in the test and qualification databases (see Clause 1.5.6), any such replication must be disclosed.

No replication was used.

Clause 2 Queries and Update Functions

Query Language

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

SQL was the query language used to implement all queries.

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.

DBGEN version 1.3.1 and QGEN version 1.3.1 were used to verify random numbers for this TPC-D benchmark.

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 1.3.1 was used to generate the substitution parameters.

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 test, 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 query text and query output.

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.

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.

Level 1 isolation was used to run the queries.

Source Code of Update Functions

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

Multiple sqlldr processes loaded data into temporary tables. Oracle 8i's parallel insert and delete functionality was used to perform the Update Functions, selecting data from the temporary tables. See Appendix E.

Database Maintenance Option

The details of the database maintenance option selected (.i.e., reset or evolve) must be disclosed (including source code of any non-commercial program used).

This implementation uses the reset option. Source code for this is included in Appendix E.

Clause 3 Database System Properties

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.

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 ORDER, LINEITEM, and HISTORY tables

1. The total price from the ORDER 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 ORDER 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.

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 ORDER, LINEITEM, and HISTORY tables.

1. The total price from the ORDER 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 ORDER 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.

Consistency

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

Consistency Test

Verify that ORDER 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 ORDER and LINEITEM tables.

1. The consistency of the ORDER and LINEITEM tables was verified based on a sample of O_ORDERKEY's.
2. 100 ACID Transactions were submitted from each of 2 execution streams.
3. The consistency of the ORDER and LINEITEM tables was reverified.

Isolation

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

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. The ACID Query was started for the same O_KEY used in step 1. The ACID Query completed and did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was COMMITTED.

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. The ACID Query was started for the same O_KEY used in step 1. The ACID Query completed and did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was ROLLED BACK.

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. The ACID Transaction 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))

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. The ACID Transaction 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.

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. The ACID Transaction was suspended prior to COMMIT.
2. Another ACID 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 ORDER, LINEITEM and HISTORY tables were changed.

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, was started which executed Q17b (from Clause 2.19) against the qualification database, with p_brand = 'Brand#23' and p_container = 'MED BOX'.
2. An ACID Transaction, T2, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. A Transaction, T3, was started which executed Q17b (from Clause 2.19) against the qualification database, with p_brand = 'Brand#23' and p_container = 'MED BOX'.
4. T1 completed.
5. T2 completed.
6. T3 competed subsequently.
7. It was verified that appropriate rows in ORDER, LINEITEM and HISTORY tables were changed.

Durability

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

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-D database tables or recovery log tables.

The disks containing TPC-D tables were RAID-0, log files were RAID-1.

The system was configured with 3 PCI bus. Two 32bit PCI bus and one 64bit PCI bus. The failure of durable medium was tested on both types of bus as followed.

1. The datafiles were backed up to an alternate disk media.
2. Two streams of ACID transactions were started.
3. A data disk was removed.
4. The two streams of ACID transactions failed and recorded their numbers of committed transactions in success files.
5. The database was brought down.
6. The datafiles were restored to their state prior to the ACID transaction streams.
7. The database ran through its recovery mode.
8. The counts in the success files and the HISTORY table count were compared. The counts matched.

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 with the loss of log test. While the test was running, one side of the mirrored set of logs was removed. After it was determined that test would still run with the loss of a log disk. Power to the server was turned off during the durability test. When power was restored, the system rebooted and the database was restarted. The durability success file and the HISTORY table were compared and the counts matched.

Memory Failure

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

See the previous section.

Clause 4 Scaling and Database Population

Ending Cardinality of Tables

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

Table	Rows
Order	150,000,000
Lineitem	600,037,902
Customer	15,000,000
Part	20,000,000
Supplier	1,000,000
Partsupp	80,000,000
Nation	25
Region	5

Distribution of Tables and Logs Across Media

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

Refer to Appendix F for details of the database layout.

Database partition/replication mapping

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

Horizontal partitioning was used for base tables LINEITEM, ORDERS, PARTS and CUSTOMER, for

materialized view logs on LINEITEM and ORDERS, for materialized views MJ_LOCS and MJ_PPSSRN and indexes L_ORED, IJ_L_ODATE, IJ_L_RDATE, IJ_L_PKEY, P_TP, P_CBP and IJ_P_NNPSPS. The details of this partitioning can be understood by examining the syntax of the table, materialized view and index definition statements in Appendix B. Similar partitioning was used in the qualification database, scaled appropriately for its size.

Data is automatically replicated in the materialized views and transparently managed in a fully ACID fashion by the database manager. Only data in the base eight tables is referenced by the seventeen queries and insert and delete operations.

The database was not replicated in total.

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	RAID0
indexes	RAID0
materialized views	RAID0
temp tablespace	RAID0
log	RAID1
usrtmp tablespace	RAID0

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 1.3.1 was used to generate the database population for this benchmark. The changes to DBGEN for partitioning of data are listed in Appendix B.

Database Content of Initial Ten Rows

The content of the first 10 rows of each table in the test database must be disclosed.

Appendix G contains the first 10 rows of each table in the test database.

Database Load Time

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

The database load time was 34 hours 16 minutes.

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.

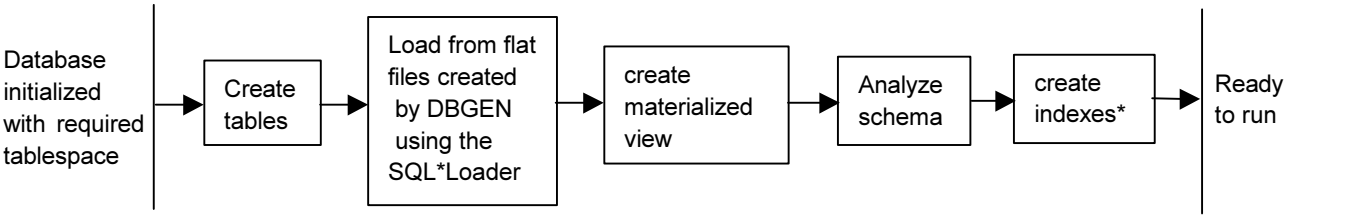
The data storage ratio is computed from the following information:

Disk type	# of Disks	Space per Disk	Sub-Total Disk Space
External 9.1GB	124	9.1GB	1128.4GB*
Internal 9.1GB	3	9.1GB	27.3GB
		Total Space	**1155.7GB***
		Data Storage Ratio	**11.6**

Database Load Mechanism Details and Illustration

The details of the database load mechanism must be described and illustrated with a block diagram.

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

Clause 5 Performance Metrics and Execution Rules

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. System Boot
2. Database Restart
3. UF1 Update Transaction
4. Stream 00 Execution
5. UF2 Update Transaction

Timing Intervals for Each Query and Update Functions

The timing intervals for each query of the measured set and for both update functions must be reported for the power test.

The power test timing intervals are contained in the Numerical Quantity Summary in the Preface of this document.

Number of Streams for the Throughput Test

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

The throughput test used only one execution stream. Since only one stream was used, the values for the power test are used for the throughput test.

Start and End Date/Times for Each Query Stream

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

The throughput test start time and finish time for each stream are contained in the Numerical Quantity Summary in the Preface of this document.

Total Elapsed Time of the Measurement Interval

The total elapsed time of the measurement interval.

The total elapsed time of the throughput test was 2,855 seconds.

Update Function Start Date/Time and Finish Date/Time

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

The start and finish time for each update function in the update stream are contained in the Numerical Quantity Summary in the Preface of this document.

Timing Intervals for Each Query and Each Update Function for Each Stream

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

The timing intervals for each query and each update function for the throughput test are contained in the Numerical Quantity Summary in the Preface of this document.

Performance Metrics

Verify that the metrics are computed as required.

The performance metrics, and the numbers on which they are based, are contained in the Numerical Quantity Summary in the Preface of this document.

Reproducibility Method

Verify that the reported metrics are repeatable.

Performance results from the two executions of the TPC-D benchmark:

Run ID	QppD@100GB	QthD@100GB	QphD@100GB
Run 1	13,554.3	2,143.6	5,390.3
Run 2	13,923.2	2,332.3	5,698.5

Clause 6 SUT and Driver Implementation

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 is performed by a shell script called audit.ksh. 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 audit_uf1.ksh script. Query submission follows, with the qexecpl.c ISL program. The execution of the UF2 script audit_uf2.ksh rounds out the Power Test execution. Both wall-clock and high-resolution times are collected for all measurement intervals.

Implementation-Specific Layer (ISL)

If any 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. QETs 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 by Clause 5.3.6.2.

Update Function

Multiple sqlldr processes loaded data into temporary tables. Oracle8i's parallel insert and delete functionality was used to perform the Update Functions, selecting data from the temporary tables.

Clause 7 Pricing

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.

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 765,474 □ (Euros). For details of pricing, see the second page of the Executive Summary.

Standard HW price include 3 years of on site maintainance, 8 hours intervention. In order to fullfill the 4 hours requirement, the maintainance price offered is quoted as a 4 hour extension of the contract, and 2 additional years of maintainance.

Software prices are from reseller price quotes. Refer to Appendix H.

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.

All hardware components used in this benchmark will be available April 30, 1999.

Oracle 8i v8.1.5.1 will be available May 18, 1999.

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. Oracle8i and WindowsNT parameters

A.1 Oracle8i parameters

```
# Init.ora include file created by bumpx.pl

parallel_adaptive_multi_user = true

parallel_max_servers      = 240
parallel_min_servers      = 120 # 30*4

pre_page_sga=true

use_indirect_data_buffers = false

query_rewrite_expression  = true

always_anti_join          = HASH
always_semi_join          = HASH
audit_trail               = FALSE
background_dump_dest      =
e:\tpcd814_work\log\hundgig
compatible                = 8.1.4.0.0
cursor_space_for_time     = FALSE

db_block_buffers          = 75000

db_block_checking         = false
db_block_checksum         = false
db_block_lru_latches      = 33
db_block_max_dirty_target = 0
db_block_size             = 8192
db_file_direct_io_count    = 96
db_file_multiblock_read_count = 96
db_files                  = 10000
db_name                   = 100g
db_writer_processes       = 4
disk_asynch_io            = true
distributed_transactions  = 0
dml_locks                 = 20000
control_files             =
('e:\control\cntr100g1','d:\control\cntr100g2')
enqueue_resources         = 64000
global_names              = FALSE
hash_area_size              =  16777216
hash_join_enabled         = true
hash_multiblock_io_count  = 64
large_pool_size           = 52428800
log_archive_start         = FALSE
log_buffer                = 5242880
log_checkpoint_interval   = 999999999999
log_checkpoints_to_alert  = true
max_dump_file_size        = 1048576
max_rollback_segments     = 1024
nls_date_format           = YYYY-MM-DD
object_cache_optimal_size = 102400
open_cursors              = 1024
optimizer_features_enable = 8.0.4
optimizer_index_caching   = 0

# 1/15 AM2:00 UPDATE
optimizer_index_cost_adj  = 25
# optimizer_index_cost_adj  = 1

optimizer_max_permutations = 80000
optimizer_mode            = CHOOSE
optimizer_percent_parallel = 100
optimizer_search_limit    = 6
parallel_broadcast_enabled = TRUE
parallel_execution_message_size = 8192
query_rewrite_enabled     = true
query_rewrite_integrity   = trusted
processes                 = 1024
remote_login_passwordfile = shared
recovery_parallelism      = 32
serializable              = FALSE
sessions                  = 1024
shared_pool_size          = 100000000

# 1/15 2:44 UPDATE
# sort_area_retained_size   =  10485760
# sort_area_size            =  10485760
sort_area_retained_size   =  12582912
sort_area_size            =  12582912

sort_multiblock_read_count = 2
sql92_security            = FALSE
transactions              = 1024
transactions_per_rollback_segment = 1
user_dump_dest            =
e:\tpcd814_work\log\hundgig
ifile                     =
e:\tpcd814_work\dout\dbs\hundgig\init_100g_defaul
t.ora
```

A.2 WindowsNT parameters

```
Microsoft Diagnostics Report For \\NT-TPCD
------------------------------------------------
---------------------

OS Version Report
------------------------------------------------
---------------------
Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 3) x86
Multiprocessor Free
Registered Owner: NEC, NEC
Product Number: 70238-415-0002856-43635
------------------------------------------------
---------------------

System Report
------------------------------------------------
---------------------
System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC
platform
BIOS Date: 12/04/98
BIOS Version: PhoenixBIOS 4.0 Release 6.0.1335

Processor list:
   0:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
   1:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
```

```
   2:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
   3:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
   4:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
   5:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
   6:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
   7:  x86 Family 6 Model 5 Stepping 3
GenuineIntel ~450 Mhz
------------------------------------------------
---------------------

Video Display Report
------------------------------------------------
---------------------
BIOS Date: 06/25/98
BIOS Version: CL-GD546x Laguna PCI VGA BIOS
Version 1.71e c15

Adapter:
   Setting: 1024 x 768 x 256
            75 Hz
   Type: cl546xm compatible display adapter
   String: Cirrus Logic VisualMedia(TM)
Accelerator
   Memory: 2 MB
   Chip Type: Cirrus Logic 5465
   DAC Type: Internal
Driver:
   Vendor: Cirrus Logic, Inc.
   File(s): cl546xm.sys, cl5465.dll
   Version: 4.00.1381.1705-1.705, 4.0.101


Drives Report
------------------------------------------------
---------------------
C:\  (Local - NTFS)  Total: 0KB, Free: 0KB
   Serial Number: D8EB - 8BA8
   Bytes per cluster: 512
   Sectors per cluster: 1
   Filename length: 255
D:\  (Local - NTFS)  Total: 8,890,348KB, Free:
2,242,092KB
   Serial Number: E4AF - E7C6
   Bytes per cluster: 512
   Sectors per cluster: 8
   Filename length: 255
E:\  (Local - NTFS)  Total: 8,890,348KB, Free:
3,921,280KB
   Serial Number: A0BC - 37A3
   Bytes per cluster: 512
   Sectors per cluster: 8
   Filename length: 255
I:\  (Local - NTFS)  Total: 124,407,788KB, Free:
11,421,600KB
   Serial Number: 8C21 - ECC3
   Bytes per cluster: 512
   Sectors per cluster: 8
   Filename length: 255

Memory Report
------------------------------------------------
---------------------
Handles: 928
Threads: 88
Processes: 13

Physical Memory (K)
   Total: 3,833,260
   Available: 3,661,588
   File Cache: 14,564

Kernel Memory (K)
   Total: 8,600
   Paged: 4,736
   Nonpaged: 3,864

Commit Charge (K)
   Total: 18,808
   Limit: 8,149,080
   Peak: 20,232

Pagefile Space (K)
   Total: 4,466,688
   Total in use: 0
   Peak: 0

   C:\pagefile.sys
      Total: 273,408
      Total in use: 0
      Peak: 0

   D:\pagefile.sys
      Total: 4,193,280
      Total in use: 0
      Peak: 0



Services Report
------------------------------------------------
---------------------
Alerter
Running   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
Computer Browser
Stopped   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
      LanmanServer
      LmHosts
ClipBook Server
Stopped   (Manual)
   C:\WINNT\system32\clipsrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      NetDDE
DHCP Client (TDI)
Stopped   (Disabled)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
```

```
   Service Dependencies:
      Tcpip
      Afd
      NetBT
EventLog (Event log)
Running   (Automatic)
   C:\WINNT\system32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Server
Stopped   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Group Dependencies:
      TDI
Workstation (NetworkProvider)
Running   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Group Dependencies:
      TDI
License Logging Service
Running   (Automatic)
   C:\WINNT\System32\llssrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
TCP/IP NetBIOS Helper
Stopped   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Group Dependencies:
      NetworkProvider
Messenger
Running   (Automatic)
   C:\WINNT\System32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
      NetBios
Network DDE (NetDDEGroup)
Stopped   (Manual)
   C:\WINNT\system32\netdde.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      NetDDEDSDM
Network DDE DSDM
Stopped   (Manual)
   C:\WINNT\system32\netdde.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Net Logon (RemoteValidation)
Stopped   (Manual)
   C:\WINNT\System32\lsass.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      LanmanWorkstation
      LmHosts
NT LM Security Support Provider
Stopped   (Manual)
   C:\WINNT\System32\SERVICES.EXE
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
OracleServicetpcd
Stopped   (Manual)
   e:\oracle814\bin\ORACLE.EXE tpcd
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
OracleTNSListenerListenertpcd
Stopped   (Manual)
   E:\oracle814\BIN\TNSLSNR
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Plug and Play (PlugPlay)
Running   (Automatic)
   C:\WINNT\system32\services.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
Remote Access Autodial Manager
Stopped   (Disabled)
   C:\WINNT\system32\rasman.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process
   Service Dependencies:
      RasMan
Remote Access Connection Manager (Network)
Stopped   (Manual)
   C:\WINNT\system32\rasman.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Shared Process, Interactive
   Service Dependencies:
      tapisrv
Remote Access Server (Network)
Stopped   (Manual)
   C:\WINNT\system32\rassrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      LanmanServer
      RasMan
      NetBios
      NetBT
Directory Replicator
Stopped   (Manual)
   C:\WINNT\System32\lmrepl.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
   Service Dependencies:
      LanmanWorkstation
      LanmanServer
Remote Procedure Call (RPC) Locator
Stopped   (Manual)
   C:\WINNT\System32\LOCATOR.EXE
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
```

```
   Service Dependencies:
      LanmanWorkstation
      Rdr
Remote Procedure Call (RPC) Service
Running   (Automatic)
   C:\WINNT\system32\RpcSs.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Schedule
Stopped   (Manual)
   C:\WINNT\System32\AtSvc.Exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
Spooler (SpoolerGroup)
Stopped   (Manual)
   C:\WINNT\system32\spoolss.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process, Interactive
Telephony Service
Stopped   (Manual)
   C:\WINNT\system32\tapisrv.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
GoodTech TELNET server for windows NT v2.1.3
Stopped   (Automatic)
   C:\telnetd\tmp\telnetd.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process
UPS
Stopped   (Manual)
   C:\WINNT\System32\ups.exe
   Service Account Name: LocalSystem
   Error Severity: Normal
   Service Flags: Own Process



Drivers Report
-----------------------------------------------
--------------------
4mmdat (SCSI Class)
Stopped   (System)
   System32\DRIVERS\4mmdat.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI Miniport
Abiosdsk (Primary disk)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
AFD Networking Support Environment (TDI)
Running   (Automatic)
   C:\WINNT\System32\drivers\afd.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Aha154x (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Aha174x (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
aic78xx (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Always (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
ami0nt (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
amsint (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Arrow (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Remote Access Mac (NDIS)
Running   (Automatic)
   C:\WINNT\system32\drivers\asyncmac.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
atapi (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Atdisk (Primary disk)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
ati (Video)
Stopped   (Disabled)
   System32\DRIVERS\ati.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Beep (Base)
Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
BusLogic (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Busmouse (Pointer Port)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cdaudio (Filter)
Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cdfs (File system)
Running   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
   Group Dependencies:
      SCSI CDROM Class
Cdrom (SCSI CDROM Class)
Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Changer (Filter)
Stopped   (System)
```

```
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
cirrus (Video)
Stopped   (Disabled)
   System32\DRIVERS\cirrus.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
cl546x (Video)
Running   (System)
   System32\DRIVERS\cl546xm.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Cpqarray (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
cpqfws2e (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
dac960nt (SCSI miniport)
Running   (Boot)
   C:\WINNT\System32\drivers\dac960nt.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Delldsa (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Dell_DGX (Video)
Stopped   (Disabled)
   System32\DRIVERS\dell_dgx.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Disk (SCSI Class)
Running   (Boot)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Diskperf (Filter)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
DptScsi (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
dtc329x (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
3Com 3C90x Adapter Driver (NDIS)
Running   (Automatic)
   C:\WINNT\System32\drivers\el90xnd4.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
et4000 (Video)
Stopped   (Disabled)
   System32\DRIVERS\et4000.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Fastfat (Boot file system)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
Fd16_700 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Fd7000ex (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Fd8xx (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
flashpnt (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Floppy (Primary disk)
Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ftdisk (Filter)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
i8042 Keyboard and PS/2 Mouse Port Driver
(Keyboard Port) Running   (System)
   System32\DRIVERS\i8042prt.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Inport (Pointer Port)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Jazzg300 (Video)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Jazzg364 (Video)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Jzvxl484 (Video)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Keyboard Class Driver (Keyboard Class)
Running   (System)
   System32\DRIVERS\kbdclass.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
KSecDD (Base)
Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
macdisk (Filter)
Running   (Boot)
   C:\WINNT\System32\drivers\macdisk.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
mainte (Extended Base)
Stopped   (Manual)
   System32\DRIVERS\mainte.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
mga (Video)
Stopped   (Disabled)
   System32\DRIVERS\mga.sys
```

```
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
mga_mil (Video)
Stopped   (Disabled)
   System32\DRIVERS\mga_mil.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
mitsumi (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
mkecr5xx (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Modem (Extended base)
Stopped   (Manual)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Mouse Class Driver (Pointer Class)
Running   (System)
   System32\DRIVERS\mouclass.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Msfs (File system)
Running   (System)
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
Mup (Network)
Running   (Manual)
   C:\WINNT\System32\drivers\mup.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
Ncr53c9x (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
ncr77c22 (Video)
Stopped   (Disabled)
   System32\DRIVERS\ncr77c22.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ncrc700 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ncrc710 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Microsoft NDIS System Driver (NDIS)
Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Microsoft NDIS TAPI driver (NDIS)
Running   (System)
   C:\WINNT\system32\drivers\ndistapi.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Remote Access WAN Wrapper (NDISWAN)
Running   (Automatic)
   C:\WINNT\system32\drivers\ndiswan.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
NetBIOS Interface (NetBIOSGroup)
Running   (Manual)
   C:\WINNT\System32\drivers\netbios.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
   Group Dependencies:
      TDI
WINS Client(TCP/IP) (PNP_TDI)
Running   (Automatic)
   C:\WINNT\System32\drivers\netbt.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      Tcpip
NetDetect
Stopped   (Manual)
   C:\WINNT\system32\drivers\netdtect.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Npfs (File system)
Running   (System)
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
Ntfs (File system)
Running   (Disabled)
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
Null (Base)
Running   (System)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Oliscsi (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Parallel (Extended base)
Stopped   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      Parport
   Group Dependencies:
      Parallel arbitrator
Parport (Parallel arbitrator)
Stopped   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
ParVdm (Extended base)
Stopped   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      Parport
   Group Dependencies:
      Parallel arbitrator
PCIDump (PCI Configuration)
Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Pcmcia (System Bus Extender)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
PnP ISA Enabler Driver (Base)
Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
psidisp (Video)
Stopped   (Disabled)
```

```
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Ql10wnt (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
qv (Video)
Stopped   (Disabled)
   System32\DRIVERS\qv.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Remote Access Auto Connection Driver (Streams
Drivers) Running   (Automatic)
   C:\WINNT\system32\drivers\rasacd.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Remote Access ARP Service (PNP_TDI)
Running   (Automatic)
   C:\WINNT\system32\drivers\rasarp.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
   Service Dependencies:
      TCPIP
Rdr (Network)
Running   (Manual)
   C:\WINNT\System32\drivers\rdr.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
s3 (Video)
Stopped   (Disabled)
   System32\DRIVERS\s3.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Scsiprnt (Extended base)
Stopped   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Scsiscan (SCSI Class)
Running   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Serial (Extended base)
Running   (Automatic)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sermouse (Pointer Port)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Sfloppy (Primary disk)
Stopped   (System)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
   Group Dependencies:
      SCSI miniport
Simbad (Filter)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
slcd32 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Sparrow (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Spock (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Srv (Network)
Running   (Manual)
   C:\WINNT\System32\drivers\srv.sys
   Error Severity: Normal
   Service Flags: File System Driver, Shared
Process
symc810 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
symc8xx (SCSI miniport)
Running   (Boot)
   C:\WINNT\system32\drivers\symc8xx.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
T128 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
T13B (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
TCP/IP Service (PNP_TDI)
Running   (Automatic)
   C:\WINNT\System32\drivers\tcpip.sys
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
tga (Video)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
tmv1 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra124 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra14f (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
Ultra24f (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
v7vram (Video)
Stopped   (Disabled)
   System32\DRIVERS\v7vram.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
VgaSave (Video Save)
Running   (System)
   C:\WINNT\System32\drivers\vga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
VgaStart (Video Init)
Stopped   (System)
   C:\WINNT\System32\drivers\vga.sys
```

```
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Wd33c93 (SCSI miniport)
Stopped   (Disabled)
   Error Severity: Normal
   Service Flags: Kernel Driver, Shared Process
wd90c24a (Video)
Stopped   (Disabled)
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
wdvga (Video)
Stopped   (Disabled)
   System32\DRIVERS\wdvga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
weitekp9 (Video)
Stopped   (Disabled)
   System32\DRIVERS\weitekp9.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process
Xga (Video)
Stopped   (Disabled)
   System32\DRIVERS\xga.sys
   Error Severity: Ignore
   Service Flags: Kernel Driver, Shared Process



IRQ and Port Report
-------------------------------------------------
---------------------
Devices                         Vector Level
Affinity
-------------------------------------------------
---------------------
MPS 1.4 - APIC platform            8     8
0x000000ff
MPS 1.4 - APIC platform            0     0
0x000000ff
MPS 1.4 - APIC platform            1     1
0x000000ff
MPS 1.4 - APIC platform            2     2
0x000000ff
MPS 1.4 - APIC platform            3     3
0x000000ff
MPS 1.4 - APIC platform            4     4
0x000000ff
MPS 1.4 - APIC platform            5     5
0x000000ff
MPS 1.4 - APIC platform            6     6
0x000000ff
MPS 1.4 - APIC platform            7     7
0x000000ff
MPS 1.4 - APIC platform            8     8
0x000000ff
MPS 1.4 - APIC platform            9     9
0x000000ff
MPS 1.4 - APIC platform           10    10
0x000000ff
MPS 1.4 - APIC platform           11    11
0x000000ff
MPS 1.4 - APIC platform           12    12
0x000000ff
MPS 1.4 - APIC platform           13    13
0x000000ff
MPS 1.4 - APIC platform           14    14
0x000000ff
MPS 1.4 - APIC platform           15    15
0x000000ff
MPS 1.4 - APIC platform           16    16
0x000000ff
MPS 1.4 - APIC platform           17    17
0x000000ff
MPS 1.4 - APIC platform           18    18
0x000000ff
MPS 1.4 - APIC platform           19    19
0x000000ff
MPS 1.4 - APIC platform           20    20
0x000000ff
MPS 1.4 - APIC platform           21    21
0x000000ff
MPS 1.4 - APIC platform           22    22
0x000000ff
MPS 1.4 - APIC platform           23    23
0x000000ff
MPS 1.4 - APIC platform           24    24
0x000000ff
MPS 1.4 - APIC platform           25    25
0x000000ff
MPS 1.4 - APIC platform           26    26
0x000000ff
MPS 1.4 - APIC platform           27    27
0x000000ff
MPS 1.4 - APIC platform           28    28
0x000000ff
MPS 1.4 - APIC platform           29    29
0x000000ff
MPS 1.4 - APIC platform           30    30
0x000000ff
MPS 1.4 - APIC platform           31    31
0x000000ff
MPS 1.4 - APIC platform           32    32
0x000000ff
MPS 1.4 - APIC platform           33    33
0x000000ff
MPS 1.4 - APIC platform           34    34
0x000000ff
MPS 1.4 - APIC platform           35    35
0x000000ff
MPS 1.4 - APIC platform           36    36
0x000000ff
MPS 1.4 - APIC platform           37    37
0x000000ff
MPS 1.4 - APIC platform           38    38
0x000000ff
MPS 1.4 - APIC platform           39    39
0x000000ff
MPS 1.4 - APIC platform           40    40
0x000000ff
MPS 1.4 - APIC platform           41    41
0x000000ff
MPS 1.4 - APIC platform           42    42
0x000000ff
MPS 1.4 - APIC platform           43    43
0x000000ff
MPS 1.4 - APIC platform           44    44
0x000000ff
MPS 1.4 - APIC platform           45    45
0x000000ff
MPS 1.4 - APIC platform           46    46
0x000000ff
MPS 1.4 - APIC platform           47    47
0x000000ff
MPS 1.4 - APIC platform           61    61
0x000000ff
MPS 1.4 - APIC platform           65    65
0x000000ff
```

```
MPS 1.4 - APIC platform           80    80
0x000000ff
MPS 1.4 - APIC platform          193   193
0x000000ff
MPS 1.4 - APIC platform          225   225
0x000000ff
MPS 1.4 - APIC platform          253   253
0x000000ff
MPS 1.4 - APIC platform          254   254
0x000000ff
MPS 1.4 - APIC platform          255   255
0x000000ff
i8042prt                           1     1
0xffffffff
i8042prt                          12    12
0xffffffff
Serial                             4     4
0x00000000
El90x                             44    44
0x00000000
Floppy                             6     6
0x00000000
dac960nt                          32    32
0x00000000
dac960nt                          32    32
0x00000000
dac960nt                          32    32
0x00000000
dac960nt                          32    32
0x00000000
dac960nt                          32    32
0x00000000
dac960nt                          32    32
0x00000000
symc8xx                           48    48
0x00000000
symc8xx                           48    48
0x00000000
-------------------------------------------------
---------------------
Devices                      Physical Address
Length
-------------------------------------------------
---------------------
MPS 1.4 - APIC platform         0x00000000
0x0000000010
MPS 1.4 - APIC platform         0x00000020
0x0000000002
MPS 1.4 - APIC platform         0x00000040
0x0000000004
MPS 1.4 - APIC platform         0x00000048
0x0000000004
MPS 1.4 - APIC platform         0x00000061
0x0000000001
MPS 1.4 - APIC platform         0x00000070
0x0000000002
MPS 1.4 - APIC platform         0x00000080
0x0000000010
MPS 1.4 - APIC platform         0x00000092
0x0000000001
MPS 1.4 - APIC platform         0x000000a0
0x0000000002
MPS 1.4 - APIC platform         0x000000c0
0x0000000010
MPS 1.4 - APIC platform         0x000000f0
0x0000000010
i8042prt                        0x00000060
0x0000000001
i8042prt                        0x00000064
0x0000000001
Serial                          0x000003f8
0x0000000007
El90x                           0x00004400
0x0000000080
Floppy                          0x000003f0
0x0000000006
Floppy                          0x000003f7
0x0000000001
dac960nt                        0x00005000
0x0000000080
dac960nt                        0x00006000
0x0000000080
dac960nt                        0x00008000
0x0000000080
dac960nt                        0x00009000
0x0000000080
dac960nt                        0x0000a000
0x0000000080
dac960nt                        0x0000b000
0x0000000080
symc8xx                         0x00004000
0x0000000100
symc8xx                         0x00007000
0x0000000100
VgaSave                         0x000003b0
0x000000000c
VgaSave                         0x000003c0
0x0000000020
VgaSave                         0x000001ce
0x0000000002
```

DMA and Memory Report

```
-------------------------------------------------
---------------------
Devices                          Channel    Port
-------------------------------------------------
---------------------
Floppy                             2     0
-------------------------------------------------
---------------------
Devices                      Physical Address
Length
-------------------------------------------------
---------------------
MPS 1.4 - APIC platform          0xfec00000
0x00000400
MPS 1.4 - APIC platform          0xfec01000
0x00000400
MPS 1.4 - APIC platform          0xfee00000
0x00000400
El90x                            0xec011000
0x00000080
dac960nt                         0xec110000
0x00000080
dac960nt                         0xee000000
0x02000000
dac960nt                         0xec210000
0x00000080
dac960nt                         0xf0000000
0x02000000
dac960nt                         0xf2210000
0x00000080
dac960nt                         0xf4000000
0x02000000
```

```
dac960nt                        0xf2310000
0x00000080
dac960nt                        0xf6000000
0x02000000
dac960nt                        0xf8110000
0x00000080
dac960nt                        0xfa000000
0x02000000
dac960nt                        0xf8210000
0x00000080
dac960nt                        0xfc000000
0x02000000
symc8xx                         0xec011400
0x00000100
symc8xx                         0xec010000
0x00001000
symc8xx                         0xf2101000
0x00000100
symc8xx                         0xf2100000
0x00001000
cl546x                          0xec000000
0x0000a000
cl546x                          0xea000000
0x02000000
VgaSave                         0x000a0000
0x00020000



Environment Report
-------------------------------------------------
---------------------



System Environment Variables
   ComSpec=C:\WINNT\system32\cmd.exe
   HOME=C:/
   NTRESKIT=C:\NTRESKIT
   NUMBER_OF_PROCESSORS=8
   OS=Windows_NT
   Os2LibPath=C:\WINNT\system32\os2\dll;

Path=d:\mks\mksnt;C:\WINNT\system32;C:\WINNT;C:\N
TRESKIT;C:\NTRESKIT\Perl;
   PROCESSOR_ARCHITECTURE=x86
   PROCESSOR_IDENTIFIER=x86 Family 6 Model 5
Stepping 3, GenuineIntel
   PROCESSOR_LEVEL=6
   PROCESSOR_REVISION=0503
   ROOTDIR=d:/mks
   SHELL=d:/mks/mksnt/sh.exe
   windir=C:\WINNT
   TMPDIR=C:/TEMP
   gt_installation_dir=C:\telnetd\tmp



Environment Variables for Current User
   DSS_CONFIG=e:\tpcd814_work\dbgenp
   DSS_PATH=i:\flatfiles
   DSS_SEED=e:\tpcd814_work\load\seeds
   ORACLE_HOME=E:\oracle814
   ORACLE_SID=tpcd
   TEMP=C:\TEMP
   TMP=C:\TEMP
   TPCD_HOME=E:\tpcd814_work
```

```
Network Report
-------------------------------------------------
---------------------
Your Access Level: Admin & Local
Workgroup or Domain: WORKGROUP
Network Version: 4.0
LanRoot: WORKGROUP
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: NT-TPCD
  Logon Server: NT-TPCD

Transport: NetBT_E190x5, 00-00-4C-D1-9F-A1, VC's:
0, Wan: Wan

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45
Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5
Datagram Reset Frequency: 60
Log Election Packets: False
Use Opportunistic Locking: True
Use Unlock Behind: True
Use Close Behind: True
Buffer Pipes: True
Use Lock, Read, Unlock: True
Use NT Caching: True
Use Raw Read: True
Use Raw Write: True
Use Write Raw Data: True
Use Encryption: True
Buffer Deny Write Files: True
Buffer Read Only Files: True
Force Core Creation: True
512 Byte Max Transfer: False
Bytes Received: 0
SMB's Received: 0
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 0
SMB's Transmitted: 0
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Initally Failed Operations: 0
Failed Completion Operations: 0
Read Operations: 0
Random Read Operations: 0
Read SMB's: 0
Large Read SMB's: 0
```

```
Small Read SMB's: 0
Write Operations: 0
Random Write Operations: 0
Write SMB's: 0
Large Write SMB's: 0
Small Write SMB's: 0
Raw Reads Denied: 0
Raw Writes Denied: 0
Network Errors: 0
Sessions: 0
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0
LM 2.x Connects: 0
Windows NT Connects: 0
Server Disconnects: 0
Hung Sessions: 0
Use Count: 0
Failed Use Count: 0
Current Commands: 0
Server File Opens: 837,580,843
Server Device Opens: 0
Server Jobs Queued: 589,824
Server Session Opens: 2
Server Sessions Timed Out: 2,147,483,736
Server Sessions Errored Out: 3
Server Password Errors: 2,147,483,776
Server Permission Errors: 4
Server System Errors: 2,147,483,856
Server Bytes Sent: 9,223,373,033,287,188,485
Server Bytes Received: 9,223,373,892,280,647,686
Server Average Response Time: 9
Server Request Buffers Needed: 2,147,484,632
Server Big Buffers Needed: 11
```

Appendix B. Programs and Scripts

B.1 ACID Source

a_q17b.sql

```
rem
rem
================================================
================+
rem        Copyright (c) 1996  Oracle Corp,
Redwood Shores, CA       |
rem                    OPEN SYSTEMS PERFORMANCE
GROUP                  |
rem                         All Rights Reserved
|
rem
================================================
================+
rem  FILENAME
rem       a_q1.sql
rem  DESCRIPTION
rem       Performs TPC-D Query 1 for TPC-D
benchmark Isolation Test 6.
rem       Asks user to input values for delta.
rem
================================================
================
rem
rem Usage:   sqlplus tpcd/tpcd @a_q1 <delta>
rem

set serverout on;

select
'BEFORE Q17B' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD
HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

select /*+ NOPARALLEL(LINEITEM) */
       sum(l_extendedprice) / 7.0 as avg_yearly
from
       lineitem,
       (select  /*+ NOPARALLEL(PARTS)
NOPARALLEL(LINEITEM) */
              p_partkey,
              0.2 * avg(l_quantity) as avgqty
         from parts, lineitem
         where
              p_partkey = l_partkey and
              p_brand = 'Brand#23' and
              p_container = 'MED BOX'
         group by p_partkey
       ) avg_quantity
where
       l_partkey = p_partkey and
       l_quantity < avgqty;

select
'AFTER Q17B' as STAGE,
substr(TO_CHAR(sysdate,'YYYY-MM-DD
HH:MI:SS'),1,20) as CURRENT_TIME
from dual;

exit;
```

a_query.sql

```
rem
rem
================================================
================+
rem        Copyright (c) 1996  Oracle Corp,
Redwood Shores, CA       |
rem                    OPEN SYSTEMS PERFORMANCE
GROUP                  |
rem                         All Rights Reserved
|
rem
================================================
================+
rem  FILENAME
rem       a_query.sql
rem  DESCRIPTION
rem       Performs ACID Query for TPC-D
benchmark.
rem       Asks user to input values for o_key
rem
================================================
================
rem
rem Usage:   sqlplus tpcd/tpcd @a_query <o_key>
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;
```

atrans.sql

```
rem
================================================
================+
rem         Copyright (c) 1995  Oracle Corp,
Redwood Shores, CA       |
rem                     OPEN SYSTEMS PERFORMANCE
GROUP                  |
rem                          All Rights Reserved
|
rem
================================================
================+
```

```
rem   FILENAME
rem        atrans.sql
rem   DESCRIPTION
rem        Creates ACID Transaction Package for
TPC-D benchmark.
rem        Asks user to input values for o_key,
delta and output file.
rem  Update: 10/24/1998       Barry N. Perkins
rem                           Added commit inside
loop so that the
rem                           exception handlers
can handle the 8177
rem                           on commit. Added
exception handler for
rem                           604, error in
recursive SQL to handle
rem                           8177 in recursive
sql.
rem
================================================
=================


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

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

CREATE OR REPLACE PACKAGE BODY d_atrans
IS
PROCEDURE doatrans
(
        l_key             IN OUT integer,
        o_key             IN OUT integer,
        delta             IN OUT integer,
        l_pkey          IN OUT integer,
        l_skey          IN OUT integer,
        l_quan          IN OUT integer,
        l_newquan       IN OUT integer,
        l_tax           IN OUT number,
        l_disc          IN OUT number,
        l_eprice        IN OUT number,
        l_neweprice     IN OUT number,
        o_tprice IN OUT number,
        o_newtprice       IN OUT number,
        rprice            IN OUT number,
        cost              IN OUT number
)
IS
        ototal number;
        not_serializable        EXCEPTION;
        not_serializable_recurs EXCEPTION;
        PRAGMA EXCEPTION_INIT(not_serializable,-
8177);
        PRAGMA
EXCEPTION_INIT(not_serializable_recurs,-604);
BEGIN
  LOOP BEGIN

    select o_totalprice
         into o_tprice
         from orders
         where o_orderkey = o_key;

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

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

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

    update orders
        set o_totalprice = o_newtprice
        where o_orderkey = o_key;

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

    EXIT;

  EXCEPTION
    WHEN not_serializable THEN
      ROLLBACK;
    WHEN not_serializable_recurs THEN
      ROLLBACK;
  END;

  END LOOP;

END doatrans;
END;
/

exit;
```

B.2 Load Process

bumpx.pl

```
#!/usr/local/bin/perl
#
# $Header: bumpx.pl
#
# bumpx.pl
#
#   Copyright (c) Oracle Corporation 1998. All
Rights Reserved.
#
#
#      bumpx.pl - BenchMark Package eXpert
#    DESCRIPTION
#           Utiltity for working with benchmark
databases
#
#
#**********************************************
*****************************
#**********************************************
*****************************
# Main Section
#**********************************************
*****************************
#**********************************************
*****************************

$os = $ENV{'OS'};
$os = "Windows_NT"; #hacked line to force os to
nt when on windows
if (($os cmp 'Windows_NT') != 0)
{                                   # os is UNIX
    $os = "unix";
}
else
{
    $os = "nt";
}

$| = 1;
$verbose=0;
$allphases                                      =
"shutdb,startdb,sdgen,dbgen,plcre,dbcre,sccre,scu
to,sctso,dapop,scuvo,ixcre,anlyz,chob,expln,query
";
if (($os cmp "unix")==0)
{
    $defphases                                  =
"sdgen,dbgen,dbcre,sctso,scuto,dapop,scuvo,ixcre,
anlyz,chob";
}
else
{
    $defphases                                  =
"sdgen,dbgen,plcre,dbcre,sctso,scuto,dapop,scuvo,
ixcre,anlyz,chob";
}
$allbmtypes = "tpcd,wisc";
$bmtype = "tpcd" if !defined $bmtype;

$pdfile = "param.txt";  # This file contains the
description of all possible parameters.
&read_parameter_description;
$runsilent = 0;

while ($arg = shift(@ARGV))
{
    if ($arg !~ /-(i|o|l|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  =  "bumpx.conf"  if  !defined
$infile;
        $filedir = $ENV{'BUMPX_DIR'};
        $infile = $filedir.$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 = "bumpx.conf" if !defined $infile;
$outfile = "bumpx.dat" if !defined $outfile;
$filedir = $ENV{'BUMPX_DIR'};
```

```
$infile = $filedir.$infile;
$outfile = $filedir.$outfile;

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);

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

## 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;
            &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;
    foreach $phase (@phases)
    {
         if ($allphases =~ /$phase/)
         {
               if ($params{'completed_phases'} !~
/$phase/)
             {
                  print "\n     Creating phase
\'$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 \'$phase\' not
built because specified in complete_phases";
             }
         }
         else
         {
               print "\n   Phase \'$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   at   "   .
localtime(time) . "." 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 $phase (@phases)
    {
        if     ($params{'completed_phases'}     !~
/$phase/)
        {
             print "\n  Building phase specific
init.ora files..." if $verbose;
            &createios();

            $phase_cmd_num = 0;

             print "\n  Removing phase specific
list file(s) (if any): $phase*.lst";
            system("rm -f $phase*.lst");

                 print   "\n    Executing   phase
\'$phase\' at " . localtime(time) 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  "\n     Completed  phase
\'$phase\' at " . localtime(time) if $verbose;
        }
        else
        {
              print  "\n   Phase  \'$phase\'  not
executed because specified in complete_phases";
        }
    }
    print  "\nExecution  pass  complete  at  "  .
localtime(time) . ".\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 cmd "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  >>
$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 =~ /\*setenv/)
                {
                    $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)
                  {
                                                $cmd     =~
s/phase\#.lst/$phase\_$phase_cmd_num.lst/;
                      ++$phase_cmd_num;
                      print TMPFILE $cmd;
                      $bgrun = $bgrun + 1;
                  }
                  else
                  {
                      close(TMPFILE);
                         system("cat  $tmpfile  >>
$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");
                                                $cmd     =~
s/phase\#.lst/$phase\_$phase_cmd_num.lst/;
                      ++$phase_cmd_num;
                      print TMPFILE $cmd;
                      $bgrun = 1;
                  }
             }
             else
             {
                  $cmd                           =~
s/phase\#.lst/$phase\_$phase_cmd_num.lst/;
                  ++$phase_cmd_num;
                  print TMPFILE $cmd;
                  close(TMPFILE);
                  system("cat     $tmpfile     >>
$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 [-s] [-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 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                                    =
$filedir.$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'})
    {
         foreach $drivenum (@drivenum)
         {
             &recreate_drive_extended_part;
         }
         if                                  (defined
$params{'plcre_log_drivenum'})
         {
                                    $drivenum      =
$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;
    }
    $d = 0;
```

```
    foreach $ts_entry (@ts_all)
    {
        &create_drive_part("$ts_entry")        if
($ts_entry =~ /ts_log/);
    }
    $d = 0;
    foreach $ts_entry (@ts_all)
    {
        &create_drive_part("$ts_entry")        if
($ts_entry !~ /ts_log/);
    }
    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 sdgen
{

&dump0("########################################
##################################");
    &dump0("#   Data   (Seed   File)   Generation
Phase");
    &time0("Begin creating seed files");

    $params{'load_tables'}                        =
$params{'tab_tables'}         if         !defined
$params{'load_tables'};
    @load_tablelist          =          split(/,/,
$params{'load_tables'});
    foreach $table (@load_tablelist)
    {
        $dbgopt = "";

        $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 .= "\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");

    $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 =~ /parts/)
         {
             $dbgtab = "P";
         }
         elsif ($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));
            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
                $fpre                           =
$params{'load_dbgen_'.$table.'_output_prefix'};
                if ($fpre =~ /\#/)
                {
                    $snum = 1;
                     $snum = (int($i/$nperset) +
1) if ($dp > 1);
                    $fpre =~ s/\#/$snum/g;
                }

                $cmd .= " -p -i ";
                $cmd                            .=
$params{'load_dbgen_'.$table.'_input_params'};
                $cmd .= sprintf(" -o %s_%d ",
$fpre,($i + 1));
            }
             $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 dbcre
{

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

# Shut down anything that may have been running
    &shutdb();  #(This will be done before eval)
    &createios();

# 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{'init_ora'}
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 public 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";
    &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
    $cmd = "";
    if ($multi)
    {
        $cmd             .=             "startup
pfile=$params{'init_ora_instance1'};\n";
    }
    else
    {
        $cmd             .=             "startup
pfile=$params{'init_ora'};\n";
    }
    &dump1("*sql");
    &dump0("*wait");

    &dump0("#   Add   remaining   datafiles   for
ts_sys");
    &add_dfs("ts_sys");
    &dump0("*wait");

    if ($params{'skip_ts'} !~ /undo/)
    {
        @ts_datafiles        =        split(/,/,
$params{'ts_undo_datafiles'});
        $ts_datafile = shift (@ts_datafiles);
        $cmd .= "create tablespace ts_undo\n";
        $cmd      .=      "             datafile
'$params{'ts_undo_area'}$ts_datafile'        size
$params{'ts_undo_first_size'}
$params{'ts_undo_options'};\n";
        &dump1("*sql");
        &dump0("*wait");
        &add_dfs("ts_undo");
         &dump0("*wait");
    }

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

    # 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";
    $cmd  .=  '@'  .  "$params{'dd_sql_area'}"  .
"utlxplan.sql;\n";
    $cmd .= "connect system/manager\n";
    $cmd  .=  '@'  .  "$params{'dd_dbs_area'}"  .
"pupbld.sql;\n";
    &dump1("*sql");
    &dump0("*wait");
    &time0("Done database creation");

# prepare for multi-user
    &recycle_db();
} # end dbcre

sub sccre
```

```
{

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

    &recycle_db()                                   if
($params{'startup_db_sccre'});

# 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");
} # end of sccre

sub sctso
{

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

         &recycle_db()                              if
($params{'startup_db_sctso'});

# 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");
        }
    }
} # end of sctso

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

&dump0("#########################################
###################################");
        &dump0("# Schema Creation Phase - User
and Tables ONLY (no datafiles)");
         &time0("Begin creating user and tables
(no datafiles)");

         &recycle_db()                          if
($params{'startup_db_scuto'});

        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");
        }
         &time0("Done  creating  user  and  tables
(no datafiles)");
    }
} # end of scuto

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

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

         &recycle_db()                          if
($params{'startup_db_scuvo'});

         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")
        }
} # end of scuvo

sub dapop
{

&dump0("########################################
##################################");
    &dump0("# Database Population 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);
    $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                                   =
$params{'load_dbgen_'.$table.'_output_prefix'};
         $pname                                    =
$params{'tab_'.$table.'_partnames'};
         $pname =~ s/\#//g;

        if ($dbgpar <= $dp)
        {
            $dpmax = $dbgpar;
            $numchild = 1;
            $remchild = 0;
        }
        else
        {
            $dpmax = $dp;
            $numchild = int($dbgpar/$dp);
              $remchild = $dbgpar - ($numchild *
$dp);
        }
        $npart             =             (defined
$params{'tab_'.$table.'_#part'} &&
                $params{'load_dbgen_partition'}  =~
/$table/) ?
            $params{'tab_'.$table.'_#part'} : 1;
        $ctlarea                                   =
$params{'load_controlfile_area'};
        $logarea                                   =
$params{'load_otherfile_area'};

        for ($ipart=1; $ipart<=$npart; $ipart++)
        {
            $idbgpar = 0;
            for ($idp=1; $idp<=$dpmax; $idp++)
            {
# create all of the "executable" load sections
                &advmulti();

                if ($dgp)
                {
                        $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)
                    {
                        $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%s%s%s%s%s%s%s%s%s%s\n",
$upwd,$control,$skip,$load,$log,$errors,$rows,
$bsize,$silent,$dirbool,$parbool);
                &dump1("*load");
                &dump0("*wait") if ((!$parbool) && (!defined $params{'load_no_waits'}));

# create the controlfiles (fixed-length fields or delimited records)
                &load_ctl_head;
                $infiles = 0;
                $badfiles = 0;
                $discardfiles = 0;
                for ($jpart=1; $jpart<=$numchild; $jpart++)
                {
                    if ($dgp)
                    {
                        @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 . "_" . $idbgpar .
                                ".dsc'";
                        }
                    }
                }
                --$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'}                 !~
/[hH][aA][sS][hH]/);
                }
                &load_ctl_tail;

            }
                  &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 ixcre
{

&dump0("########################################
####################################");
    &dump0("# Index Creation Phase");

    &recycle_db()                                   if
($params{'startup_db_ixcre'});

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

         $cmd              .=              "connect
$params{'user'}/$params{'passwd'};\n";
        if   ($params{'con_'.$const.'_constraint'}
!~ /null/)
        {
            $cmd       .=       "alter       table
$params{'con_'.$const.'_table'} " .
                "drop constraint $const;\n";
        }
        &dump1("*sql");
        &time0("Done      dropping      constraint
$const");
    }
    &dump0("*wait") if (($os cmp "nt") ==0);
    &time0("Done dropping constraints");
    &time0("Begin dropping indices");

    foreach $index (@indexlist)
    {
        &time0("Begin dropping index $index");
         $cmd              .=              "connect
$params{'user'}/$params{'passwd'};\n";
        $cmd .= "drop index $index;\n";
        $cmd       .=       "alter       tablespace
$params{'ind_'.$index.'_ts'}    coalesce;\n"    if
((defined     $params{'ind_'.$index.'_ts'})     &&
($params{'skip_ts'} !~ /index/));
        &dump1("*sql");
        &time0("Done dropping index $index");
    }
    &dump0("*wait") if (($os cmp "nt") ==0);
    &time0("Done dropping indices");

    &time0("Recycling    database    to    ensure
freespace cleanup");
    &recycle_db();

    &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";
        $cmd .= "create ";
        if  ($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'} ";
            $cmd .= "add constraint $const\n";
            $cmd                 .=               "
$params{'con_'.$const.'_constraint'}           key
($params{'con_'.$const.'_columns'})\n";
                                                  if
($params{'con_'.$const.'_has_index'}           !~
/[Tt][Rr][Uu][Ee]/) {
                $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'};
                if   ($params{'compatible'}   =~
/7\./)
                {
                                      $cmd    .=    "
unrecoverable\n"                                  if
($params{'con_'.$const.'_unrecoverable'}       =~
/[tT][rR][uU][eE]/);
                } elsif  ($params{'compatible'}
=~ /8\./) {
                    $cmd .= "      nologging\n"
if       ($params{'con_'.$const.'_nolg'}       =~
/[tT][rR][uU][eE]/);
                }
                $cmd  .=  "             compute
statistics\n"                                     if
($params{'con_'.$const.'_compstats'}           =~
/[tT][rR][uU][eE]/);
```

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

            }
                                                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'} ";
                $cmd    .=    "enable    primary
key;\n";
            }
        }
        &dump1("*sql");
         &dump0("*wait") if (($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_1
{
    @ind_partnames = ();
    $params{'ind_'.$index.'_#part'} =
         $params{$ob_stype.'_'.$params{'ind_'.$in
dex.'_'.$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].'_no
lg'})
            {
                $cmd .= "nologging\n"
                                                  if
($params{'ind_'.$index.'_'.$ind_partnames[$i].'_n
olg'}
                          =~ /[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
',$index,$ind_partnames[$i]);
          &process_part_params('ind','%u','pctused
',$index,$ind_partnames[$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.'_'.$ind_partnames
[$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].'_no
lg'})
         {
             $cmd .= "nologging\n"
                  if
($params{'ind_'.$index.'_'.$ind_partnames[$i].'_n
olg'}
                      =~ /[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.'_partva
ls'});
          }
        }

        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_for
mat'})                                          ?
$params{$ob_stype.'_'.$indtab.'_'.$col.'_date_for
mat'} : "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");

    &recycle_db()                                 if
($params{'startup_db_anlyz'});

    if   ($params{'anl_type'}   =~   /via   package
dbms.stats/)
    {
        &time0("Begin   via   package   dbms.stats
analyzing");
        $cmd             =             "connect
$params{'user'}/$params{'passwd'};\n";
        foreach $object (@anlyzlist)
        {
            $phead="anl_".$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'}.'_#pa
rt'};
                    }
                        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.'_partnames'}        .
$nextfile;
                        }
                    }
                        @ind_partnames    =
split(/,/,$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.'_partnames'}        .
$filenm .
                            $addcomma;
                            $filenm   =
$savename;
                        }
        @tab_partnames=split(/,/,$params{$ind_en
try.'_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");

     &recycle_db()                                  if
($params{'startup_db_chob'});

     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.'_degpa
r'} =~ /[dD][eE][fF][aA][uU][lL][tT]/)
            {
```

```
                $cmd .= " parallel";
            }
                                                elsif
($params{'chob_'.$objecttypes.'_'.$object.'_degpa
r'}                                                 =~
/[nN][oO][pP][aA][rR][aA][lL][lL][eE][lL]/)
            {
                $cmd      .=      "      parallel
$params{'chob_'.$objecttypes.'_'.$object.'_degpar
'}"
                    if                      (defined
$params{'chob_'.$objecttypes.'_'.$object.'_degpar
'});
            }
            $cmd .= ";\n";
        }
        if                                  (defined
$params{'chob_'.$objecttypes.'_'.$object.'_storag
e'})
        {
            $cmd .= "alter $objecttype $objectn";
            if                              (defined
$params{'chob_'.$objecttypes.'_'.$object.'_storag
e'})
            {
                $cmd      .=      "      storage
($params{'chob_'.$objecttypes.'_'.$object.'_stora
ge'});\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");

    &recycle_db()                               if
($params{'startup_db_expln'});

    $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");

    &recycle_db()                                if
($params{'startup_db_query'});

    $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 recycle_db
{
    &createios();
    &time0("Cycling    database    before    phase
\'$phase\'...\n") if $verbose;
    &time0("Shutting   down   database...\n")   if
$verbose;
    &shutdb();
    &time0("Starting database...\n") if $verbose;
    &startdb();
} # end of recycle_db

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

sub shutdb
{
# shutdown whatever might be running
    if ($multi)
    {
        for ($instance = $num_inst; $instance >=
1; $instance--)
        {
            &setinst($instance);
            $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;\n";
                $cmd   .=   "create   tablespace
$tsname temporary\n";
            }
            else          #      user      managed
permanent tablespace
            {
                $cmd    .=    "drop    tablespace
$tsname;\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;\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]
/))
                {
                    $cmd .= "autoallocate\n";
                }
                if                        (defined
($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;\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]
/))
            {
                $cmd .= "autoallocate\n";
            }
                                  if      (defined
($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_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++)
    {
        $ts_datafile = shift (@ts_datafiles);
        &advmulti();
         if ($tsname =~ /ts_sys/)
         {
            $cmd .= "alter tablespace system\n";
         }
         else
         {
            $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, "always_anti_join=HASH");
    push (@iokv, "always_semi_join=HASH");
    push (@iokv, "audit_trail=FALSE");
    push                                    (@iokv,
"background_dump_dest=$params{'dbs_trace'}");
```

```
    push (@iokv, "compatible=8.1.4.0.0");
    push (@iokv, "cursor_space_for_time=FALSE");
    push      (@iokv,      "db_block_buffers=38400
#300m");
    push (@iokv, "db_block_checking=false");
    push (@iokv, "db_block_checksum=false");
    push (@iokv, "db_block_lru_latches=24");
    push (@iokv, "db_block_max_dirty_target=0");
    push (@iokv, "db_block_size=8192");
    push                                    (@iokv,
"_db_file_noncontig_mblock_read_count=11");
    push                                    (@iokv,
"db_file_multiblock_read_count=64");
    push (@iokv, "db_files=10000");
    push                                    (@iokv,
"db_name=$params{'oracle_sid'}");
    push (@iokv, "db_writer_processes=4");
    push (@iokv, "disk_asynch_io=true");
    push (@iokv, "distributed_transactions=0");
    push (@iokv, "dml_locks=20000");
    push                                    (@iokv,
"control_files=$params{'io_control_files'}");
    push (@iokv, "enqueue_resources=64000");
    push (@iokv, "global_names=FALSE");
    push (@iokv, "hash_area_size=20971520 #20m");
    push (@iokv, "hash_join_enabled=true");
    push (@iokv, "hash_multiblock_io_count=64");
    push (@iokv, "log_archive_start=FALSE");
    push (@iokv, "log_buffer=5242880");
    push                                    (@iokv,
"log_checkpoint_interval=999999999999");
    push                                    (@iokv,
"log_checkpoints_to_alert=true");
    push (@iokv, "max_dump_file_size=1048576");
    push (@iokv, "max_rollback_segments=1024");
    push (@iokv, "nls_date_format=YYYY-MM-DD");
    push                                    (@iokv,
"object_cache_optimal_size=102400");
    push (@iokv, "open_cursors=1024");
    push                                    (@iokv,
"optimizer_features_enable=8.0.4");
    push (@iokv, "optimizer_index_caching=0");
    push (@iokv, "optimizer_index_cost_adj=25");
    push                                    (@iokv,
"optimizer_max_permutations=80000");
    push (@iokv, "optimizer_mode=CHOOSE");
    push                                    (@iokv,
"optimizer_percent_parallel=100");
    push (@iokv, "optimizer_search_limit=6");
    push                                    (@iokv,
"parallel_broadcast_enabled=TRUE");
    push                                    (@iokv,
"parallel_execution_message_size=4096");
    push (@iokv, "parallel_max_servers=48");
    push (@iokv, "parallel_min_servers=0");
    push (@iokv, "query_rewrite_enabled=true");
    push                                    (@iokv,
"query_rewrite_integrity=trusted");
    push (@iokv, "processes=1024");
    push                                    (@iokv,
"remote_login_passwordfile=shared");
    push (@iokv, "recovery_parallelism=12");
    push (@iokv, "serializable=FALSE");
    push (@iokv, "sessions=1024");
    push    (@iokv,    "shared_pool_size=200000000
#200m");
    push                                    (@iokv,
"sort_area_retained_size=10485760 #10m");
    push (@iokv, "sort_area_size=10485760 #10m");
    push (@iokv, "sort_multiblock_read_count=2");
#                               push       (@iokv,
"sort_multiblock_read_count=8");
    push (@iokv, "sql92_security=FALSE");
    push (@iokv, "transactions=1024");
    push                                    (@iokv,
"transactions_per_rollback_segment=1");
    push                                    (@iokv,
"user_dump_dest=$params{'dbs_trace'}");

    if ($params{'skip_mk_initoras'} !~ /include/)
    {
        open (IOFILE, ">$params{'init_ora'}");
        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;
        }
         $entry   =   sprintf   ("%-25s   =   %s\n",
"ifile", $params{'startupfile_'.$phase});
         print IOFILE $entry;
        close (IOFILE);

         if (!-e $params{'startupfile_'.$phase})
         {
                                  open      (IOFILE,
">$params{'startupfile_'.$phase}");
              print  IOFILE  "#  Init.ora  include
file created by bumpx.pl\n";
             print IOFILE "# This file name was
specified in the config file\n";
             print IOFILE "# but didn't exist. An
empty one was created.\n\n";
            close (IOFILE);
             print "\nINIT.ORA file created: " .
$params{'startupfile_'.$phase}    .    "\n"    if
$verbose;
         }
    }

    if ($multi)
    {
         for   ($instance   =   1;   $instance   <=
$num_inst; $instance++)
         {
               if  ($params{'skip_mk_initoras'}  !~
/instance/)
             {
                  open                    (IOFILE,
">$params{'init_ora_instance'.$instance}");
                  print  IOFILE  "#  Init.ora  for
instance $instance created by bumpx.pl\n\n";
                  print    IOFILE    "thread    =
$instance\n"
                                                  if
($params{'special_machine'} ne 'ncube');
                  print    IOFILE    "ifile    =
$params{'init_ora'}\n";
                  close (IOFILE);
             }
         }
    }
}

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_select'});
        $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_select'});
    }
    $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/,/,\n/g;
    $as_select =~ s/from/\nFROM\n/g;
    $as_select =~ s/select/SELECT\n/g;
    $as_select =~ s/where/\nWHERE\n/g;
    $as_select =~ s/group by/\nGROUP BY\n/g;
    $as_select =~ s/ and /\n AND \n/g;
    $as_select =~ s/ 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,$o
b_type);
    }
    elsif    ($params{$ob_entry.'_parttype'}    =~
/[hH][aA][sS][hH]/)
    {
        $cmd  .=  "partition  by  hash  ("  .
$params{$ob_entry.'_partcol'} . ")\n(\n";
        &expand_partitions_doit($ob_entry,$ob,$o
b_type);
    }
    elsif    ($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,$o
b_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

    if     ($params{$ob_entry.'_parttype'}     !~
/[hH][aA][sS][hH]/)
    {
        &process_part_brys ($ob_entry,$ob);
    }
    &process_part_ts ($ob_entry,$ob);

    # complete the partition statement

    for ($i=0; $i < $params{$ob_entry.'_#part'};
$i++)
    {
        if   ($params{$ob_entry.'_parttype'}   !~
/[hH][aA][sS][hH]/)
        {
             $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";
            }
        }
        else
        {
             $cmd   .=   "partition   "   .
```

```
$ob_partnames[$i] . "\n";
        }
        &process_part_params($ob_type,'%f','pctf
ree',$ob,$ob_partnames[$i]);
        &process_part_params($ob_type,'it','init
rans',$ob,$ob_partnames[$i]);
        &process_part_params($ob_type,'mt','maxt
rans',$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'})
    {
        @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'} = "";
                    for   ($i   =   1;   $i   <=
$params{$ob__entry.'_#part'}; $i++)
            {
                $filenm =~ s/\#/$i/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_forma
t'} : "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/ || $file =~
/^cntr/) ? $params{'plcre_log_drivenum'} :
        @drivenum[$d];
    $cmd = "creapart -l PhysicalDrive";
    $cmd .= $drivenum . " " . $psize . "\n";
    &dump1("*sh");
    $cmd = $file . "\t\\device\\harddisk" .
$drivenum;
    $cmd .=       "\\partition"                 .
++$partnum{$drivenum} . "\n";
    &time0($cmd);
    print LNKSFILE $cmd;
    if ($file !~/^log/)
    {
        $d = ($d < $md) && ($d < $numfiles) ?
++$d : 0;
    }
}

sub load_ctl_head
{
    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);
}

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;
}

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";
    }
}

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

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

    if (defined $params{'compatible'})
    {
        $pre72 = 1 if $params{'compatible'} =~
/7\.(0|1)/;
        $pre73 = 1 if $params{'compatible'} =~
/7\.(0|1|2)/;
        $pre80 = 1 if $params{'compatible'} =~
/7\.(0|1|2|3)/;
        $pre81 = 1 if $params{'compatible'} =~
/8\.(0)/;
    }

    $verbose     =     $params{'verbose'}     =~
/[tT][rR][uU][eE]/ ? 1 : 0;
    $debug       =      $params{'debug'}       =~
/[tT][rR][uU][eE]/ ? 1 : 0;

    $params{'dbs_area'}                         =~
s/\?/$ENV{'ORACLE_HOME'}/g;
    $params{'dbs_trace'}                        =~
s/\?/$ENV{'ORACLE_HOME'}/g;
    $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{'init_ora'}   =   sprintf   ("%s%s",
$params{'dbs_area'},
"init_$params{'oracle_sid'}.ora");

    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'};
         $params{'startup_db_'.$phase} = (defined
$params{'startup_db_'.$phase}) ?
            $params{'startup_db_'.$phase} : 0;
        if                                  (defined
$params{'startupfile_'.$phase})
        {
                $params{'startupfile_'.$phase} =
$params{'dbs_area'}       .       "init_"       .
$params{'oracle_sid'} .
                 "_"                                .
$params{'startupfile_'.$phase} . ".ora";
        }
        elsif                               (defined
$params{'startupfile_default'})
        {
                $params{'startupfile_'.$phase} =
$params{'dbs_area'}       .       "init_"       .
$params{'oracle_sid'} .
                 "_"                                .
$params{'startupfile_default'} . ".ora";
        }
        else
        {
                $params{'startupfile_'.$phase} =
$params{'dbs_area'}       .       "init_"       .
$params{'oracle_sid'} .
                 "_" . $phase . ".ora";
        }
```

```
        $params{'startupfile_'.$phase} = "/host"
. $params{'startupfile_'.$phase}
            if    ($params{'special_machine'}    eq
'ncube');

        for   ($instance   =   1;   $instance   <=
$params{$phase.'_num_inst'}; $instance++)
        {

$params{'init_ora_instance'.$instance} =
                sprintf                ("%s%s",
$params{'dbs_area'},
"init".$j."_$params{'oracle_sid'}.ora");

$params{'init_ora_instance'.$instance} =
                sprintf             ("/host%s",
$params{'init_ora_instance'.$instance})
                if   ($params{'special_machine'}
eq 'ncube');
        }


    }
    $params{'ts_undo_#rs'}                      =
$params{'def_num_inst'}        if        !defined
$params{'ts_undo_#rs'};
    $params{'ts_log_#files'}                    =
$params{'ts_log_#threads'}                      *
$params{'ts_log_#files_pt'};

    $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'};
            }
        }
    }

    $mostts                                     =
"$params{'ts_temp_names'},$params{'ts_data_names'
},$params{'ts_index_names'}";
    $allts                                      =
"ts_sys,ts_log,ts_undo,$params{'ts_temp_names'},$
params{'ts_data_names'},$params{'ts_index_names'}
";
    @ts_data              =              split(/,/,
$params{'ts_data_names'});
    @ts_index             =              split(/,/,
$params{'ts_index_names'});
    @ts_temp              =              split(/,/,
$params{'ts_temp_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'};
    }

    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)
    {
        $params{'tab_'.$table.'_ts'}  =  "ts_data"
if !defined $params{'tab_'.$table.'_ts'};
        $params{'tab_'.$table.'_load_degpar'}    =
$params{'load_deg_parallel'}
                                if      !defined
$params{'tab_'.$table.'_load_degpar'};
        $params{'tab_'.$table.'_#rows'}         *=
$params{'scale_factor'};
        $params{'load_dbgen_'.$table.'_option_C'}
= $params{'load_dbgen_def_option_C'}
                                if      !defined
$params{'load_dbgen_'.$table.'_option_C'};

$params{'load_dbgen_'.$table.'input_params'}     =
$params{'load_dbgen_def_input_params'}
                                if      !defined
$params{'load_dbgen_'.$table.'_input_params'};

$params{'load_dbgen_'.$table.'output_prefix'}    =
$params{'load_flatfile_area'}
                                if      !defined
$params{'load_dbgen_'.$table.'_output_prefix'};
    }

    @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'};
         $params{'view_'.$view.'_rewrite'}  =  ""
if !defined $params{'view_'.$view.'_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/));
    }
    $params{'anl_type'}     =     "via     package
dbms.stats" if !defined($params{'anl_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'};
    }
}

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=normal');
    push (@defparams, 'def_num_inst=1');
#                           push      (@defparams,
'privileges=resource,unlimited
tablespace,connect');
    push    (@defparams,    'privileges=DBA,query
rewrite,global query rewrite');
    push (@defparams, 'qry_number_trials=1');
}

sub tpcd
{
    @defparams = ();
    push (@defparams, 'user=tpcd');
    push (@defparams, 'passwd=tpcd');
    push                                  (@defparams,
'tab_tables=lineitem,orders,partsupp,parts,custom
er,supplier,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_return
flag,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_supp
key,l_linenumber,l_quantity,l_extendedprice,l_dis
count,l_tax,l_returnflag,l_linestatus,l_shipdate,
l_commitdate,l_receiptdate,l_shipinstruct,l_shipm
ode,l_comment');
    push                                  (@defparams,
'tab_orders_columns=o_orderkey,o_custkey,o_orders
tatus,o_totalprice,o_orderdate,o_orderpriority,o_
clerk,o_shippriority,o_comment');
    push                                  (@defparams,
'tab_partsupp_columns=ps_partkey,ps_suppkey,ps_av
ailqty,ps_supplycost,ps_comment');
    push                                  (@defparams,
'tab_parts_columns=p_partkey,p_name,p_mfgr,p_bran
d,p_type,p_size,p_container,p_retailprice,p_comme
nt');
    push                                  (@defparams,
'tab_customer_columns=c_custkey,c_name,c_address,
c_nationkey,c_phone,c_acctbal,c_mktsegment,c_comm
ent');
    push                                  (@defparams,
'tab_supplier_columns=s_suppkey,s_name,s_address,
s_nationkey,s_phone,s_acctbal,s_comment');
    push                                  (@defparams,
'tab_nation_columns=n_nationkey,n_name,n_regionke
y,n_comment');
    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_parts_#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=varchar(25)');
    push                                (@defparams,
'tab_lineitem_l_shipmode_type=varchar(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=varchar(15)');
    push                                (@defparams,
'tab_orders_o_clerk_type=varchar(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_parts_parttype=range');
    push                                (@defparams,
'tab_parts_p_partkey_type=number');
    push                                (@defparams,
'tab_parts_p_name_type=varchar(55)');
    push                                (@defparams,
'tab_parts_p_mfgr_type=varchar(25)');
    push                                (@defparams,
'tab_parts_p_brand_type=varchar(10)');
    push                                (@defparams,
'tab_parts_p_type_type=varchar(25)');
    push                                (@defparams,
```

```
'tab_parts_p_size_type=number');
    push                                (@defparams,
'tab_parts_p_container_type=varchar(10)');
    push                                (@defparams,
'tab_parts_p_retailprice_type=number');
    push                                (@defparams,
'tab_parts_p_comment_type=varchar(23)');
    push                                (@defparams,
'tab_parts_p_partkey_pos=1:12');
    push                                (@defparams,
'tab_parts_p_name_pos=13:67');
    push                                (@defparams,
'tab_parts_p_mfgr_pos=68:92');
    push                                (@defparams,
'tab_parts_p_brand_pos=93:102');
    push                                (@defparams,
'tab_parts_p_type_pos=103:127');
    push                                (@defparams,
'tab_parts_p_size_pos=128:139');
    push                                (@defparams,
'tab_parts_p_container_pos=140:149');
    push                                (@defparams,
'tab_parts_p_retailprice_pos=150:164');
    push                                (@defparams,
'tab_parts_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=varchar(15)');
    push                                (@defparams,
'tab_customer_c_acctbal_type=number');
    push                                (@defparams,
'tab_customer_c_mktsegment_type=varchar(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=varchar(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=varchar(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=varchar(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=varchar(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='.$params{'loa
d_flatfile_area'}.'lineitem');
    push                                (@defparams,
'load_dbgen_orders_output_prefix='.$params{'load_
flatfile_area'}.'orders');
    push                                (@defparams,
'load_dbgen_customer_output_prefix='.$params{'loa
d_flatfile_area'}.'customer');
    push                                (@defparams,
'load_dbgen_parts_output_prefix='.$params{'load_f
latfile_area'}.'parts');
    push                                (@defparams,
'load_dbgen_partsupp_output_prefix='.$params{'loa
d_flatfile_area'}.'partsupp');
    push                                (@defparams,
'load_dbgen_supplier_output_prefix='.$params{'loa
d_flatfile_area'}.'supplier');
    push                                (@defparams,
```

```
'load_dbgen_nation_output_prefix='.$params{'load_
flatfile_area'}.'nation');
    push                                (@defparams,
'load_dbgen_region_output_prefix='.$params{'load_
flatfile_area'}.'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,tw
enty,hundred,thousand,fivethous,tenthous,odd100,e
ven100,stringu1,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 time0
{
    if (($os cmp "nt") == 0)
        {
            $value = shift (@_);
            $value = "*time=" . $value;
            $value = $value . "\n" if ($value !~
/.*\n$/);
            print OUTFILE "$value";
            $value = "";
        }
}
```

100g_816_anlyz.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-anlyz
*bgon=1
################################################
############################
# Analyze Phase
# *time=Begin anlyze
# *time=Begin via package dbms.stats analyzing
```

```
*sql
{
connect tpcd/tpcd;
execute dbms_stats.gather_schema_stats('tpcd' ,
estimate_percent => 1 , degree => 8 , granularity
=> 'GLOBAL' , block_sample => true);
}
*wait
*sql
{
connect tpcd/tpcd;
execute dbms_stats.gather_schema_stats('sys' ,
estimate_percent=>10);
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_lineitem');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_orders');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_customer');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_supplier');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_parts');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_partsupp');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_nation');
}
*sql
{
connect tpcd/tpcd;
execute dbms_stats.delete_table_stats('tpcd' ,
'MLOG$_region');
}
# *time=End per schema analyzing
*wait
*bgoff
%e-anlyz
```

100g_816_dapop.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dapop
*bgon=30
################################################
############################
# Database Population Phase
*time=Begin data population
*time=Cycling database before phase 'dapop'...
*time=Shutting down database...
*sql
{
shutdown normal;
}
*wait
*time=Starting database...
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
}
*wait
*time=Begin lineitem load
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_1.ctl
e:\tpcd814_work\load\log\lineitem_1_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_1.ctl
e:\tpcd814_work\load\log\lineitem_2_1.log
direct=true parallel=true
```

```
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_1.ctl
e:\tpcd814_work\load\log\lineitem_3_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_1.ctl
e:\tpcd814_work\load\log\lineitem_4_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_1.ctl
e:\tpcd814_work\load\log\lineitem_5_1.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 1 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_2.ctl
e:\tpcd814_work\load\log\lineitem_1_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_2.ctl
e:\tpcd814_work\load\log\lineitem_2_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_2.ctl
e:\tpcd814_work\load\log\lineitem_3_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_2.ctl
e:\tpcd814_work\load\log\lineitem_4_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_2.ctl
e:\tpcd814_work\load\log\lineitem_5_2.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 2 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_3.ctl
e:\tpcd814_work\load\log\lineitem_1_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_3.ctl
e:\tpcd814_work\load\log\lineitem_2_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_3.ctl
e:\tpcd814_work\load\log\lineitem_3_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_3.ctl
e:\tpcd814_work\load\log\lineitem_4_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_3.ctl
e:\tpcd814_work\load\log\lineitem_5_3.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 3 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_4.ctl
e:\tpcd814_work\load\log\lineitem_1_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_4.ctl
e:\tpcd814_work\load\log\lineitem_2_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_4.ctl
e:\tpcd814_work\load\log\lineitem_3_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_4.ctl
e:\tpcd814_work\load\log\lineitem_4_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_4.ctl
e:\tpcd814_work\load\log\lineitem_5_4.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 4 for Table:
```

```
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_5.ctl
e:\tpcd814_work\load\log\lineitem_1_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_5.ctl
e:\tpcd814_work\load\log\lineitem_2_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_5.ctl
e:\tpcd814_work\load\log\lineitem_3_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_5.ctl
e:\tpcd814_work\load\log\lineitem_4_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_5.ctl
e:\tpcd814_work\load\log\lineitem_5_5.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 5 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_6.ctl
e:\tpcd814_work\load\log\lineitem_1_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_6.ctl
e:\tpcd814_work\load\log\lineitem_2_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_6.ctl
e:\tpcd814_work\load\log\lineitem_3_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_6.ctl
e:\tpcd814_work\load\log\lineitem_4_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_6.ctl
e:\tpcd814_work\load\log\lineitem_5_6.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 6 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_7.ctl
e:\tpcd814_work\load\log\lineitem_1_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_7.ctl
e:\tpcd814_work\load\log\lineitem_2_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_7.ctl
e:\tpcd814_work\load\log\lineitem_3_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_7.ctl
e:\tpcd814_work\load\log\lineitem_4_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_7.ctl
e:\tpcd814_work\load\log\lineitem_5_7.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 7 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_8.ctl
e:\tpcd814_work\load\log\lineitem_1_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_8.ctl
e:\tpcd814_work\load\log\lineitem_2_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_8.ctl
e:\tpcd814_work\load\log\lineitem_3_8.log
direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_8.ctl
e:\tpcd814_work\load\log\lineitem_4_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_8.ctl
e:\tpcd814_work\load\log\lineitem_5_8.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 8 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_9.ctl
e:\tpcd814_work\load\log\lineitem_1_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_9.ctl
e:\tpcd814_work\load\log\lineitem_2_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_9.ctl
e:\tpcd814_work\load\log\lineitem_3_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_9.ctl
e:\tpcd814_work\load\log\lineitem_4_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_9.ctl
e:\tpcd814_work\load\log\lineitem_5_9.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 9 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_10.ctl
e:\tpcd814_work\load\log\lineitem_1_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_10.ctl
e:\tpcd814_work\load\log\lineitem_2_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_10.ctl
e:\tpcd814_work\load\log\lineitem_3_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_10.ctl
e:\tpcd814_work\load\log\lineitem_4_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_10.ctl
e:\tpcd814_work\load\log\lineitem_5_10.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 10 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_11.ctl
e:\tpcd814_work\load\log\lineitem_1_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_11.ctl
e:\tpcd814_work\load\log\lineitem_2_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_11.ctl
e:\tpcd814_work\load\log\lineitem_3_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_11.ctl
e:\tpcd814_work\load\log\lineitem_4_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_11.ctl
e:\tpcd814_work\load\log\lineitem_5_11.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 11 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_12.ctl
e:\tpcd814_work\load\log\lineitem_1_12.log
direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_12.ctl
e:\tpcd814_work\load\log\lineitem_2_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_12.ctl
e:\tpcd814_work\load\log\lineitem_3_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_12.ctl
e:\tpcd814_work\load\log\lineitem_4_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_12.ctl
e:\tpcd814_work\load\log\lineitem_5_12.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 12 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_13.ctl
e:\tpcd814_work\load\log\lineitem_1_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_13.ctl
e:\tpcd814_work\load\log\lineitem_2_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_13.ctl
e:\tpcd814_work\load\log\lineitem_3_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_13.ctl
e:\tpcd814_work\load\log\lineitem_4_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_13.ctl
e:\tpcd814_work\load\log\lineitem_5_13.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 13 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_14.ctl
e:\tpcd814_work\load\log\lineitem_1_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_14.ctl
e:\tpcd814_work\load\log\lineitem_2_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_14.ctl
e:\tpcd814_work\load\log\lineitem_3_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_14.ctl
e:\tpcd814_work\load\log\lineitem_4_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_14.ctl
e:\tpcd814_work\load\log\lineitem_5_14.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 14 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_15.ctl
e:\tpcd814_work\load\log\lineitem_1_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_15.ctl
e:\tpcd814_work\load\log\lineitem_2_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_15.ctl
e:\tpcd814_work\load\log\lineitem_3_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_15.ctl
e:\tpcd814_work\load\log\lineitem_4_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_15.ctl
e:\tpcd814_work\load\log\lineitem_5_15.log
```

```
direct=true parallel=true
}
*wait
*time=End of load for Partition: 15 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_16.ctl
e:\tpcd814_work\load\log\lineitem_1_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_16.ctl
e:\tpcd814_work\load\log\lineitem_2_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_16.ctl
e:\tpcd814_work\load\log\lineitem_3_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_16.ctl
e:\tpcd814_work\load\log\lineitem_4_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_16.ctl
e:\tpcd814_work\load\log\lineitem_5_16.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 16 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_17.ctl
e:\tpcd814_work\load\log\lineitem_1_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_17.ctl
e:\tpcd814_work\load\log\lineitem_2_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_17.ctl
e:\tpcd814_work\load\log\lineitem_3_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_17.ctl
e:\tpcd814_work\load\log\lineitem_4_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_17.ctl
e:\tpcd814_work\load\log\lineitem_5_17.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 17 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_18.ctl
e:\tpcd814_work\load\log\lineitem_1_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_18.ctl
e:\tpcd814_work\load\log\lineitem_2_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_18.ctl
e:\tpcd814_work\load\log\lineitem_3_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_18.ctl
e:\tpcd814_work\load\log\lineitem_4_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_18.ctl
e:\tpcd814_work\load\log\lineitem_5_18.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 18 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_19.ctl
e:\tpcd814_work\load\log\lineitem_1_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_19.ctl
e:\tpcd814_work\load\log\lineitem_2_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_19.ctl
e:\tpcd814_work\load\log\lineitem_3_19.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_19.ctl
e:\tpcd814_work\load\log\lineitem_4_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_19.ctl
e:\tpcd814_work\load\log\lineitem_5_19.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 19 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_20.ctl
e:\tpcd814_work\load\log\lineitem_1_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_20.ctl
e:\tpcd814_work\load\log\lineitem_2_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_20.ctl
e:\tpcd814_work\load\log\lineitem_3_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_20.ctl
e:\tpcd814_work\load\log\lineitem_4_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_20.ctl
e:\tpcd814_work\load\log\lineitem_5_20.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 20 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_21.ctl
e:\tpcd814_work\load\log\lineitem_1_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_21.ctl
e:\tpcd814_work\load\log\lineitem_2_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_21.ctl
e:\tpcd814_work\load\log\lineitem_3_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_21.ctl
e:\tpcd814_work\load\log\lineitem_4_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_21.ctl
e:\tpcd814_work\load\log\lineitem_5_21.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 21 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_22.ctl
e:\tpcd814_work\load\log\lineitem_1_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_22.ctl
e:\tpcd814_work\load\log\lineitem_2_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_22.ctl
e:\tpcd814_work\load\log\lineitem_3_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_22.ctl
e:\tpcd814_work\load\log\lineitem_4_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_22.ctl
e:\tpcd814_work\load\log\lineitem_5_22.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 22 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_23.ctl
e:\tpcd814_work\load\log\lineitem_1_23.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_23.ctl
e:\tpcd814_work\load\log\lineitem_2_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_23.ctl
e:\tpcd814_work\load\log\lineitem_3_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_23.ctl
e:\tpcd814_work\load\log\lineitem_4_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_23.ctl
e:\tpcd814_work\load\log\lineitem_5_23.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 23 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_24.ctl
e:\tpcd814_work\load\log\lineitem_1_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_24.ctl
e:\tpcd814_work\load\log\lineitem_2_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_24.ctl
e:\tpcd814_work\load\log\lineitem_3_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_24.ctl
e:\tpcd814_work\load\log\lineitem_4_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_24.ctl
e:\tpcd814_work\load\log\lineitem_5_24.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 24 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_25.ctl
e:\tpcd814_work\load\log\lineitem_1_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_25.ctl
e:\tpcd814_work\load\log\lineitem_2_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_25.ctl
e:\tpcd814_work\load\log\lineitem_3_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_25.ctl
e:\tpcd814_work\load\log\lineitem_4_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_25.ctl
e:\tpcd814_work\load\log\lineitem_5_25.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 25 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_26.ctl
e:\tpcd814_work\load\log\lineitem_1_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_26.ctl
e:\tpcd814_work\load\log\lineitem_2_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_26.ctl
e:\tpcd814_work\load\log\lineitem_3_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_26.ctl
e:\tpcd814_work\load\log\lineitem_4_26.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_26.ctl
e:\tpcd814_work\load\log\lineitem_5_26.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 26 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_27.ctl
e:\tpcd814_work\load\log\lineitem_1_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_27.ctl
e:\tpcd814_work\load\log\lineitem_2_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_27.ctl
e:\tpcd814_work\load\log\lineitem_3_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_27.ctl
e:\tpcd814_work\load\log\lineitem_4_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_27.ctl
e:\tpcd814_work\load\log\lineitem_5_27.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 27 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_28.ctl
e:\tpcd814_work\load\log\lineitem_1_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_28.ctl
e:\tpcd814_work\load\log\lineitem_2_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_28.ctl
e:\tpcd814_work\load\log\lineitem_3_28.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_28.ctl
e:\tpcd814_work\load\log\lineitem_4_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_28.ctl
e:\tpcd814_work\load\log\lineitem_5_28.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 28 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_29.ctl
e:\tpcd814_work\load\log\lineitem_1_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_29.ctl
e:\tpcd814_work\load\log\lineitem_2_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_29.ctl
e:\tpcd814_work\load\log\lineitem_3_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_29.ctl
e:\tpcd814_work\load\log\lineitem_4_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_29.ctl
e:\tpcd814_work\load\log\lineitem_5_29.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 29 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_30.ctl
e:\tpcd814_work\load\log\lineitem_1_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_30.ctl
e:\tpcd814_work\load\log\lineitem_2_30.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_30.ctl
e:\tpcd814_work\load\log\lineitem_3_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_30.ctl
e:\tpcd814_work\load\log\lineitem_4_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_30.ctl
e:\tpcd814_work\load\log\lineitem_5_30.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 30 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_31.ctl
e:\tpcd814_work\load\log\lineitem_1_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_31.ctl
e:\tpcd814_work\load\log\lineitem_2_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_31.ctl
e:\tpcd814_work\load\log\lineitem_3_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_31.ctl
e:\tpcd814_work\load\log\lineitem_4_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_31.ctl
e:\tpcd814_work\load\log\lineitem_5_31.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 31 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_32.ctl
e:\tpcd814_work\load\log\lineitem_1_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_32.ctl
e:\tpcd814_work\load\log\lineitem_2_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_32.ctl
e:\tpcd814_work\load\log\lineitem_3_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_32.ctl
e:\tpcd814_work\load\log\lineitem_4_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_32.ctl
e:\tpcd814_work\load\log\lineitem_5_32.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 32 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_33.ctl
e:\tpcd814_work\load\log\lineitem_1_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_33.ctl
e:\tpcd814_work\load\log\lineitem_2_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_33.ctl
e:\tpcd814_work\load\log\lineitem_3_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_33.ctl
e:\tpcd814_work\load\log\lineitem_4_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_33.ctl
e:\tpcd814_work\load\log\lineitem_5_33.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 33 for Table:
lineitem
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_34.ctl
e:\tpcd814_work\load\log\lineitem_1_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_34.ctl
e:\tpcd814_work\load\log\lineitem_2_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_34.ctl
e:\tpcd814_work\load\log\lineitem_3_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_34.ctl
e:\tpcd814_work\load\log\lineitem_4_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_34.ctl
e:\tpcd814_work\load\log\lineitem_5_34.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 34 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_35.ctl
e:\tpcd814_work\load\log\lineitem_1_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_35.ctl
e:\tpcd814_work\load\log\lineitem_2_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_35.ctl
e:\tpcd814_work\load\log\lineitem_3_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_35.ctl
e:\tpcd814_work\load\log\lineitem_4_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_35.ctl
e:\tpcd814_work\load\log\lineitem_5_35.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 35 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_36.ctl
e:\tpcd814_work\load\log\lineitem_1_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_36.ctl
e:\tpcd814_work\load\log\lineitem_2_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_36.ctl
e:\tpcd814_work\load\log\lineitem_3_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_36.ctl
e:\tpcd814_work\load\log\lineitem_4_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_36.ctl
e:\tpcd814_work\load\log\lineitem_5_36.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 36 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_37.ctl
e:\tpcd814_work\load\log\lineitem_1_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_37.ctl
e:\tpcd814_work\load\log\lineitem_2_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_37.ctl
e:\tpcd814_work\load\log\lineitem_3_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_37.ctl
```

```
e:\tpcd814_work\load\log\lineitem_4_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_37.ctl
e:\tpcd814_work\load\log\lineitem_5_37.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 37 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_38.ctl
e:\tpcd814_work\load\log\lineitem_1_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_38.ctl
e:\tpcd814_work\load\log\lineitem_2_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_38.ctl
e:\tpcd814_work\load\log\lineitem_3_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_38.ctl
e:\tpcd814_work\load\log\lineitem_4_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_38.ctl
e:\tpcd814_work\load\log\lineitem_5_38.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 38 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_39.ctl
e:\tpcd814_work\load\log\lineitem_1_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_39.ctl
e:\tpcd814_work\load\log\lineitem_2_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_39.ctl
e:\tpcd814_work\load\log\lineitem_3_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_39.ctl
e:\tpcd814_work\load\log\lineitem_4_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_39.ctl
e:\tpcd814_work\load\log\lineitem_5_39.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 39 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_40.ctl
e:\tpcd814_work\load\log\lineitem_1_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_40.ctl
e:\tpcd814_work\load\log\lineitem_2_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_40.ctl
e:\tpcd814_work\load\log\lineitem_3_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_40.ctl
e:\tpcd814_work\load\log\lineitem_4_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_40.ctl
e:\tpcd814_work\load\log\lineitem_5_40.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 40 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_41.ctl
e:\tpcd814_work\load\log\lineitem_1_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_41.ctl
```

```
e:\tpcd814_work\load\log\lineitem_2_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_41.ctl
e:\tpcd814_work\load\log\lineitem_3_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_41.ctl
e:\tpcd814_work\load\log\lineitem_4_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_41.ctl
e:\tpcd814_work\load\log\lineitem_5_41.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 41 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_42.ctl
e:\tpcd814_work\load\log\lineitem_1_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_42.ctl
e:\tpcd814_work\load\log\lineitem_2_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_42.ctl
e:\tpcd814_work\load\log\lineitem_3_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_42.ctl
e:\tpcd814_work\load\log\lineitem_4_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_42.ctl
e:\tpcd814_work\load\log\lineitem_5_42.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 42 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_43.ctl
e:\tpcd814_work\load\log\lineitem_1_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_43.ctl
e:\tpcd814_work\load\log\lineitem_2_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_43.ctl
e:\tpcd814_work\load\log\lineitem_3_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_43.ctl
e:\tpcd814_work\load\log\lineitem_4_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_43.ctl
e:\tpcd814_work\load\log\lineitem_5_43.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 43 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_44.ctl
e:\tpcd814_work\load\log\lineitem_1_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_44.ctl
e:\tpcd814_work\load\log\lineitem_2_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_44.ctl
e:\tpcd814_work\load\log\lineitem_3_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_44.ctl
e:\tpcd814_work\load\log\lineitem_4_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_44.ctl
e:\tpcd814_work\load\log\lineitem_5_44.log
direct=true parallel=true
}
```

```
*wait
*time=End of load for Partition: 44 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_45.ctl
e:\tpcd814_work\load\log\lineitem_1_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_45.ctl
e:\tpcd814_work\load\log\lineitem_2_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_45.ctl
e:\tpcd814_work\load\log\lineitem_3_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_45.ctl
e:\tpcd814_work\load\log\lineitem_4_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_45.ctl
e:\tpcd814_work\load\log\lineitem_5_45.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 45 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_46.ctl
e:\tpcd814_work\load\log\lineitem_1_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_46.ctl
e:\tpcd814_work\load\log\lineitem_2_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_46.ctl
e:\tpcd814_work\load\log\lineitem_3_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_46.ctl
e:\tpcd814_work\load\log\lineitem_4_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_46.ctl
e:\tpcd814_work\load\log\lineitem_5_46.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 46 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_47.ctl
e:\tpcd814_work\load\log\lineitem_1_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_47.ctl
e:\tpcd814_work\load\log\lineitem_2_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_47.ctl
e:\tpcd814_work\load\log\lineitem_3_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_47.ctl
e:\tpcd814_work\load\log\lineitem_4_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_47.ctl
e:\tpcd814_work\load\log\lineitem_5_47.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 47 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_48.ctl
e:\tpcd814_work\load\log\lineitem_1_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_48.ctl
e:\tpcd814_work\load\log\lineitem_2_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_48.ctl
e:\tpcd814_work\load\log\lineitem_3_48.log
direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_48.ctl
e:\tpcd814_work\load\log\lineitem_4_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_48.ctl
e:\tpcd814_work\load\log\lineitem_5_48.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 48 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_49.ctl
e:\tpcd814_work\load\log\lineitem_1_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_49.ctl
e:\tpcd814_work\load\log\lineitem_2_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_49.ctl
e:\tpcd814_work\load\log\lineitem_3_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_49.ctl
e:\tpcd814_work\load\log\lineitem_4_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_49.ctl
e:\tpcd814_work\load\log\lineitem_5_49.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 49 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_50.ctl
e:\tpcd814_work\load\log\lineitem_1_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_50.ctl
e:\tpcd814_work\load\log\lineitem_2_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_50.ctl
e:\tpcd814_work\load\log\lineitem_3_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_50.ctl
e:\tpcd814_work\load\log\lineitem_4_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_50.ctl
e:\tpcd814_work\load\log\lineitem_5_50.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 50 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_51.ctl
e:\tpcd814_work\load\log\lineitem_1_51.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_51.ctl
e:\tpcd814_work\load\log\lineitem_2_51.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_51.ctl
e:\tpcd814_work\load\log\lineitem_3_51.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_51.ctl
e:\tpcd814_work\load\log\lineitem_4_51.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_51.ctl
e:\tpcd814_work\load\log\lineitem_5_51.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 51 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_52.ctl
e:\tpcd814_work\load\log\lineitem_1_52.log
direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_52.ctl
e:\tpcd814_work\load\log\lineitem_2_52.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_52.ctl
e:\tpcd814_work\load\log\lineitem_3_52.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_52.ctl
e:\tpcd814_work\load\log\lineitem_4_52.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_52.ctl
e:\tpcd814_work\load\log\lineitem_5_52.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 52 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_53.ctl
e:\tpcd814_work\load\log\lineitem_1_53.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_53.ctl
e:\tpcd814_work\load\log\lineitem_2_53.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_53.ctl
e:\tpcd814_work\load\log\lineitem_3_53.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_53.ctl
e:\tpcd814_work\load\log\lineitem_4_53.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_53.ctl
e:\tpcd814_work\load\log\lineitem_5_53.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 53 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_54.ctl
e:\tpcd814_work\load\log\lineitem_1_54.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_54.ctl
e:\tpcd814_work\load\log\lineitem_2_54.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_54.ctl
e:\tpcd814_work\load\log\lineitem_3_54.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_54.ctl
e:\tpcd814_work\load\log\lineitem_4_54.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_54.ctl
e:\tpcd814_work\load\log\lineitem_5_54.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 54 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_55.ctl
e:\tpcd814_work\load\log\lineitem_1_55.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_55.ctl
e:\tpcd814_work\load\log\lineitem_2_55.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_55.ctl
e:\tpcd814_work\load\log\lineitem_3_55.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_55.ctl
e:\tpcd814_work\load\log\lineitem_4_55.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\lineitem_5_55.ctl
e:\tpcd814_work\load\log\lineitem_5_55.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 55 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_56.ctl
e:\tpcd814_work\load\log\lineitem_1_56.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_56.ctl
e:\tpcd814_work\load\log\lineitem_2_56.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_56.ctl
e:\tpcd814_work\load\log\lineitem_3_56.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_56.ctl
e:\tpcd814_work\load\log\lineitem_4_56.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_56.ctl
e:\tpcd814_work\load\log\lineitem_5_56.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 56 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_57.ctl
e:\tpcd814_work\load\log\lineitem_1_57.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_57.ctl
e:\tpcd814_work\load\log\lineitem_2_57.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_57.ctl
e:\tpcd814_work\load\log\lineitem_3_57.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_57.ctl
e:\tpcd814_work\load\log\lineitem_4_57.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_57.ctl
e:\tpcd814_work\load\log\lineitem_5_57.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 57 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_58.ctl
e:\tpcd814_work\load\log\lineitem_1_58.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_58.ctl
e:\tpcd814_work\load\log\lineitem_2_58.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_58.ctl
e:\tpcd814_work\load\log\lineitem_3_58.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_58.ctl
e:\tpcd814_work\load\log\lineitem_4_58.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_58.ctl
e:\tpcd814_work\load\log\lineitem_5_58.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 58 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_59.ctl
e:\tpcd814_work\load\log\lineitem_1_59.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_59.ctl
e:\tpcd814_work\load\log\lineitem_2_59.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\lineitem_3_59.ctl
e:\tpcd814_work\load\log\lineitem_3_59.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_59.ctl
e:\tpcd814_work\load\log\lineitem_4_59.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_59.ctl
e:\tpcd814_work\load\log\lineitem_5_59.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 59 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_60.ctl
e:\tpcd814_work\load\log\lineitem_1_60.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_60.ctl
e:\tpcd814_work\load\log\lineitem_2_60.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_60.ctl
e:\tpcd814_work\load\log\lineitem_3_60.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_60.ctl
e:\tpcd814_work\load\log\lineitem_4_60.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_60.ctl
e:\tpcd814_work\load\log\lineitem_5_60.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 60 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_61.ctl
e:\tpcd814_work\load\log\lineitem_1_61.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_61.ctl
e:\tpcd814_work\load\log\lineitem_2_61.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_61.ctl
e:\tpcd814_work\load\log\lineitem_3_61.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_61.ctl
e:\tpcd814_work\load\log\lineitem_4_61.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_61.ctl
e:\tpcd814_work\load\log\lineitem_5_61.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 61 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_62.ctl
e:\tpcd814_work\load\log\lineitem_1_62.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_62.ctl
e:\tpcd814_work\load\log\lineitem_2_62.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_62.ctl
e:\tpcd814_work\load\log\lineitem_3_62.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_62.ctl
e:\tpcd814_work\load\log\lineitem_4_62.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_62.ctl
e:\tpcd814_work\load\log\lineitem_5_62.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 62 for Table:
lineitem
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\lineitem_1_63.ctl
e:\tpcd814_work\load\log\lineitem_1_63.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_63.ctl
e:\tpcd814_work\load\log\lineitem_2_63.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_63.ctl
e:\tpcd814_work\load\log\lineitem_3_63.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_63.ctl
e:\tpcd814_work\load\log\lineitem_4_63.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_63.ctl
e:\tpcd814_work\load\log\lineitem_5_63.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 63 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_64.ctl
e:\tpcd814_work\load\log\lineitem_1_64.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_64.ctl
e:\tpcd814_work\load\log\lineitem_2_64.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_64.ctl
e:\tpcd814_work\load\log\lineitem_3_64.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_64.ctl
e:\tpcd814_work\load\log\lineitem_4_64.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_64.ctl
e:\tpcd814_work\load\log\lineitem_5_64.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 64 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_65.ctl
e:\tpcd814_work\load\log\lineitem_1_65.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_65.ctl
e:\tpcd814_work\load\log\lineitem_2_65.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_65.ctl
e:\tpcd814_work\load\log\lineitem_3_65.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_65.ctl
e:\tpcd814_work\load\log\lineitem_4_65.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_65.ctl
e:\tpcd814_work\load\log\lineitem_5_65.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 65 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_66.ctl
e:\tpcd814_work\load\log\lineitem_1_66.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_66.ctl
e:\tpcd814_work\load\log\lineitem_2_66.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_66.ctl
e:\tpcd814_work\load\log\lineitem_3_66.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_66.ctl
e:\tpcd814_work\load\log\lineitem_4_66.log
direct=true parallel=true
```

```
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_66.ctl
e:\tpcd814_work\load\log\lineitem_5_66.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 66 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_67.ctl
e:\tpcd814_work\load\log\lineitem_1_67.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_67.ctl
e:\tpcd814_work\load\log\lineitem_2_67.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_67.ctl
e:\tpcd814_work\load\log\lineitem_3_67.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_67.ctl
e:\tpcd814_work\load\log\lineitem_4_67.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_67.ctl
e:\tpcd814_work\load\log\lineitem_5_67.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 67 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_68.ctl
e:\tpcd814_work\load\log\lineitem_1_68.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_68.ctl
e:\tpcd814_work\load\log\lineitem_2_68.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_68.ctl
e:\tpcd814_work\load\log\lineitem_3_68.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_68.ctl
e:\tpcd814_work\load\log\lineitem_4_68.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_68.ctl
e:\tpcd814_work\load\log\lineitem_5_68.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 68 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_69.ctl
e:\tpcd814_work\load\log\lineitem_1_69.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_69.ctl
e:\tpcd814_work\load\log\lineitem_2_69.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_69.ctl
e:\tpcd814_work\load\log\lineitem_3_69.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_69.ctl
e:\tpcd814_work\load\log\lineitem_4_69.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_69.ctl
e:\tpcd814_work\load\log\lineitem_5_69.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 69 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_70.ctl
e:\tpcd814_work\load\log\lineitem_1_70.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_70.ctl
e:\tpcd814_work\load\log\lineitem_2_70.log
direct=true parallel=true
```

```
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_70.ctl
e:\tpcd814_work\load\log\lineitem_3_70.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_70.ctl
e:\tpcd814_work\load\log\lineitem_4_70.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_70.ctl
e:\tpcd814_work\load\log\lineitem_5_70.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 70 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_71.ctl
e:\tpcd814_work\load\log\lineitem_1_71.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_71.ctl
e:\tpcd814_work\load\log\lineitem_2_71.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_71.ctl
e:\tpcd814_work\load\log\lineitem_3_71.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_71.ctl
e:\tpcd814_work\load\log\lineitem_4_71.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_71.ctl
e:\tpcd814_work\load\log\lineitem_5_71.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 71 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_72.ctl
e:\tpcd814_work\load\log\lineitem_1_72.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_72.ctl
e:\tpcd814_work\load\log\lineitem_2_72.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_72.ctl
e:\tpcd814_work\load\log\lineitem_3_72.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_72.ctl
e:\tpcd814_work\load\log\lineitem_4_72.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_72.ctl
e:\tpcd814_work\load\log\lineitem_5_72.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 72 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_73.ctl
e:\tpcd814_work\load\log\lineitem_1_73.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_73.ctl
e:\tpcd814_work\load\log\lineitem_2_73.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_73.ctl
e:\tpcd814_work\load\log\lineitem_3_73.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_73.ctl
e:\tpcd814_work\load\log\lineitem_4_73.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_73.ctl
e:\tpcd814_work\load\log\lineitem_5_73.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 73 for Table:
```

```
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_74.ctl
e:\tpcd814_work\load\log\lineitem_1_74.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_74.ctl
e:\tpcd814_work\load\log\lineitem_2_74.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_74.ctl
e:\tpcd814_work\load\log\lineitem_3_74.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_74.ctl
e:\tpcd814_work\load\log\lineitem_4_74.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_74.ctl
e:\tpcd814_work\load\log\lineitem_5_74.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 74 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_75.ctl
e:\tpcd814_work\load\log\lineitem_1_75.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_75.ctl
e:\tpcd814_work\load\log\lineitem_2_75.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_75.ctl
e:\tpcd814_work\load\log\lineitem_3_75.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_75.ctl
e:\tpcd814_work\load\log\lineitem_4_75.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_75.ctl
e:\tpcd814_work\load\log\lineitem_5_75.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 75 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_76.ctl
e:\tpcd814_work\load\log\lineitem_1_76.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_76.ctl
e:\tpcd814_work\load\log\lineitem_2_76.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_76.ctl
e:\tpcd814_work\load\log\lineitem_3_76.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_76.ctl
e:\tpcd814_work\load\log\lineitem_4_76.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_76.ctl
e:\tpcd814_work\load\log\lineitem_5_76.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 76 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_77.ctl
e:\tpcd814_work\load\log\lineitem_1_77.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_77.ctl
e:\tpcd814_work\load\log\lineitem_2_77.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_77.ctl
e:\tpcd814_work\load\log\lineitem_3_77.log
direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_77.ctl
e:\tpcd814_work\load\log\lineitem_4_77.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_77.ctl
e:\tpcd814_work\load\log\lineitem_5_77.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 77 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_78.ctl
e:\tpcd814_work\load\log\lineitem_1_78.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_78.ctl
e:\tpcd814_work\load\log\lineitem_2_78.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_78.ctl
e:\tpcd814_work\load\log\lineitem_3_78.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_78.ctl
e:\tpcd814_work\load\log\lineitem_4_78.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_78.ctl
e:\tpcd814_work\load\log\lineitem_5_78.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 78 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_79.ctl
e:\tpcd814_work\load\log\lineitem_1_79.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_79.ctl
e:\tpcd814_work\load\log\lineitem_2_79.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_79.ctl
e:\tpcd814_work\load\log\lineitem_3_79.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_79.ctl
e:\tpcd814_work\load\log\lineitem_4_79.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_79.ctl
e:\tpcd814_work\load\log\lineitem_5_79.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 79 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_80.ctl
e:\tpcd814_work\load\log\lineitem_1_80.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_80.ctl
e:\tpcd814_work\load\log\lineitem_2_80.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_80.ctl
e:\tpcd814_work\load\log\lineitem_3_80.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_80.ctl
e:\tpcd814_work\load\log\lineitem_4_80.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_80.ctl
e:\tpcd814_work\load\log\lineitem_5_80.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 80 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_81.ctl
e:\tpcd814_work\load\log\lineitem_1_81.log
direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_81.ctl
e:\tpcd814_work\load\log\lineitem_2_81.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_81.ctl
e:\tpcd814_work\load\log\lineitem_3_81.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_81.ctl
e:\tpcd814_work\load\log\lineitem_4_81.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_81.ctl
e:\tpcd814_work\load\log\lineitem_5_81.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 81 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_82.ctl
e:\tpcd814_work\load\log\lineitem_1_82.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_82.ctl
e:\tpcd814_work\load\log\lineitem_2_82.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_82.ctl
e:\tpcd814_work\load\log\lineitem_3_82.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_82.ctl
e:\tpcd814_work\load\log\lineitem_4_82.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_82.ctl
e:\tpcd814_work\load\log\lineitem_5_82.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 82 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_83.ctl
e:\tpcd814_work\load\log\lineitem_1_83.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_83.ctl
e:\tpcd814_work\load\log\lineitem_2_83.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_83.ctl
e:\tpcd814_work\load\log\lineitem_3_83.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_83.ctl
e:\tpcd814_work\load\log\lineitem_4_83.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_83.ctl
e:\tpcd814_work\load\log\lineitem_5_83.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 83 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_84.ctl
e:\tpcd814_work\load\log\lineitem_1_84.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_84.ctl
e:\tpcd814_work\load\log\lineitem_2_84.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_84.ctl
e:\tpcd814_work\load\log\lineitem_3_84.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_84.ctl
e:\tpcd814_work\load\log\lineitem_4_84.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_84.ctl
e:\tpcd814_work\load\log\lineitem_5_84.log
```

```
direct=true parallel=true
}
*wait
*time=End of load for Partition: 84 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_85.ctl
e:\tpcd814_work\load\log\lineitem_1_85.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_85.ctl
e:\tpcd814_work\load\log\lineitem_2_85.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_85.ctl
e:\tpcd814_work\load\log\lineitem_3_85.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_85.ctl
e:\tpcd814_work\load\log\lineitem_4_85.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_85.ctl
e:\tpcd814_work\load\log\lineitem_5_85.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 85 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_86.ctl
e:\tpcd814_work\load\log\lineitem_1_86.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_86.ctl
e:\tpcd814_work\load\log\lineitem_2_86.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_86.ctl
e:\tpcd814_work\load\log\lineitem_3_86.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_86.ctl
e:\tpcd814_work\load\log\lineitem_4_86.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_86.ctl
e:\tpcd814_work\load\log\lineitem_5_86.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 86 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_87.ctl
e:\tpcd814_work\load\log\lineitem_1_87.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_87.ctl
e:\tpcd814_work\load\log\lineitem_2_87.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_87.ctl
e:\tpcd814_work\load\log\lineitem_3_87.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_87.ctl
e:\tpcd814_work\load\log\lineitem_4_87.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_87.ctl
e:\tpcd814_work\load\log\lineitem_5_87.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 87 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_88.ctl
e:\tpcd814_work\load\log\lineitem_1_88.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_88.ctl
e:\tpcd814_work\load\log\lineitem_2_88.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_88.ctl
e:\tpcd814_work\load\log\lineitem_3_88.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_88.ctl
e:\tpcd814_work\load\log\lineitem_4_88.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_88.ctl
e:\tpcd814_work\load\log\lineitem_5_88.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 88 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_89.ctl
e:\tpcd814_work\load\log\lineitem_1_89.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_89.ctl
e:\tpcd814_work\load\log\lineitem_2_89.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_89.ctl
e:\tpcd814_work\load\log\lineitem_3_89.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_89.ctl
e:\tpcd814_work\load\log\lineitem_4_89.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_89.ctl
e:\tpcd814_work\load\log\lineitem_5_89.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 89 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_90.ctl
e:\tpcd814_work\load\log\lineitem_1_90.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_90.ctl
e:\tpcd814_work\load\log\lineitem_2_90.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_90.ctl
e:\tpcd814_work\load\log\lineitem_3_90.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_90.ctl
e:\tpcd814_work\load\log\lineitem_4_90.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_90.ctl
e:\tpcd814_work\load\log\lineitem_5_90.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 90 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_91.ctl
e:\tpcd814_work\load\log\lineitem_1_91.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_91.ctl
e:\tpcd814_work\load\log\lineitem_2_91.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_91.ctl
e:\tpcd814_work\load\log\lineitem_3_91.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_91.ctl
e:\tpcd814_work\load\log\lineitem_4_91.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_91.ctl
e:\tpcd814_work\load\log\lineitem_5_91.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 91 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_92.ctl
e:\tpcd814_work\load\log\lineitem_1_92.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_92.ctl
e:\tpcd814_work\load\log\lineitem_2_92.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_92.ctl
e:\tpcd814_work\load\log\lineitem_3_92.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_92.ctl
e:\tpcd814_work\load\log\lineitem_4_92.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_92.ctl
e:\tpcd814_work\load\log\lineitem_5_92.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 92 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_93.ctl
e:\tpcd814_work\load\log\lineitem_1_93.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_93.ctl
e:\tpcd814_work\load\log\lineitem_2_93.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_93.ctl
e:\tpcd814_work\load\log\lineitem_3_93.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_93.ctl
e:\tpcd814_work\load\log\lineitem_4_93.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_93.ctl
e:\tpcd814_work\load\log\lineitem_5_93.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 93 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_94.ctl
e:\tpcd814_work\load\log\lineitem_1_94.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_94.ctl
e:\tpcd814_work\load\log\lineitem_2_94.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_94.ctl
e:\tpcd814_work\load\log\lineitem_3_94.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_94.ctl
e:\tpcd814_work\load\log\lineitem_4_94.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_94.ctl
e:\tpcd814_work\load\log\lineitem_5_94.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 94 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_95.ctl
e:\tpcd814_work\load\log\lineitem_1_95.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_95.ctl
e:\tpcd814_work\load\log\lineitem_2_95.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_95.ctl
e:\tpcd814_work\load\log\lineitem_3_95.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_95.ctl
e:\tpcd814_work\load\log\lineitem_4_95.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_95.ctl
e:\tpcd814_work\load\log\lineitem_5_95.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 95 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_96.ctl
e:\tpcd814_work\load\log\lineitem_1_96.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_96.ctl
e:\tpcd814_work\load\log\lineitem_2_96.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_96.ctl
e:\tpcd814_work\load\log\lineitem_3_96.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_96.ctl
e:\tpcd814_work\load\log\lineitem_4_96.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_96.ctl
e:\tpcd814_work\load\log\lineitem_5_96.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 96 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_97.ctl
e:\tpcd814_work\load\log\lineitem_1_97.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_97.ctl
e:\tpcd814_work\load\log\lineitem_2_97.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_97.ctl
e:\tpcd814_work\load\log\lineitem_3_97.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_97.ctl
e:\tpcd814_work\load\log\lineitem_4_97.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_97.ctl
e:\tpcd814_work\load\log\lineitem_5_97.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 97 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_98.ctl
e:\tpcd814_work\load\log\lineitem_1_98.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_98.ctl
e:\tpcd814_work\load\log\lineitem_2_98.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_98.ctl
e:\tpcd814_work\load\log\lineitem_3_98.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_98.ctl
e:\tpcd814_work\load\log\lineitem_4_98.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_98.ctl
e:\tpcd814_work\load\log\lineitem_5_98.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 98 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_99.ctl
e:\tpcd814_work\load\log\lineitem_1_99.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_99.ctl
e:\tpcd814_work\load\log\lineitem_2_99.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_99.ctl
e:\tpcd814_work\load\log\lineitem_3_99.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_99.ctl
e:\tpcd814_work\load\log\lineitem_4_99.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_99.ctl
e:\tpcd814_work\load\log\lineitem_5_99.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 99 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_100.ctl
e:\tpcd814_work\load\log\lineitem_1_100.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_100.ctl
e:\tpcd814_work\load\log\lineitem_2_100.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_100.ctl
e:\tpcd814_work\load\log\lineitem_3_100.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_100.ctl
e:\tpcd814_work\load\log\lineitem_4_100.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_100.ctl
e:\tpcd814_work\load\log\lineitem_5_100.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 100 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_101.ctl
e:\tpcd814_work\load\log\lineitem_1_101.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_101.ctl
e:\tpcd814_work\load\log\lineitem_2_101.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_101.ctl
e:\tpcd814_work\load\log\lineitem_3_101.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_101.ctl
e:\tpcd814_work\load\log\lineitem_4_101.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_101.ctl
e:\tpcd814_work\load\log\lineitem_5_101.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 101 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_102.ctl
e:\tpcd814_work\load\log\lineitem_1_102.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_102.ctl
e:\tpcd814_work\load\log\lineitem_2_102.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_102.ctl
e:\tpcd814_work\load\log\lineitem_3_102.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_102.ctl
e:\tpcd814_work\load\log\lineitem_4_102.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_102.ctl
e:\tpcd814_work\load\log\lineitem_5_102.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 102 for Table:
lineitem
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_103.ctl
e:\tpcd814_work\load\log\lineitem_1_103.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_103.ctl
e:\tpcd814_work\load\log\lineitem_2_103.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_103.ctl
e:\tpcd814_work\load\log\lineitem_3_103.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_103.ctl
e:\tpcd814_work\load\log\lineitem_4_103.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_103.ctl
e:\tpcd814_work\load\log\lineitem_5_103.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 103 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_104.ctl
e:\tpcd814_work\load\log\lineitem_1_104.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_104.ctl
e:\tpcd814_work\load\log\lineitem_2_104.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_104.ctl
e:\tpcd814_work\load\log\lineitem_3_104.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_104.ctl
e:\tpcd814_work\load\log\lineitem_4_104.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_104.ctl
e:\tpcd814_work\load\log\lineitem_5_104.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 104 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_105.ctl
e:\tpcd814_work\load\log\lineitem_1_105.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_105.ctl
e:\tpcd814_work\load\log\lineitem_2_105.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_105.ctl
e:\tpcd814_work\load\log\lineitem_3_105.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_105.ctl
e:\tpcd814_work\load\log\lineitem_4_105.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_105.ctl
e:\tpcd814_work\load\log\lineitem_5_105.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 105 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_106.ctl
e:\tpcd814_work\load\log\lineitem_1_106.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_106.ctl
e:\tpcd814_work\load\log\lineitem_2_106.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_106.ctl
e:\tpcd814_work\load\log\lineitem_3_106.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_106.ctl
```

```
e:\tpcd814_work\load\log\lineitem_4_106.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_106.ctl
e:\tpcd814_work\load\log\lineitem_5_106.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 106 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_107.ctl
e:\tpcd814_work\load\log\lineitem_1_107.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_107.ctl
e:\tpcd814_work\load\log\lineitem_2_107.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_107.ctl
e:\tpcd814_work\load\log\lineitem_3_107.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_107.ctl
e:\tpcd814_work\load\log\lineitem_4_107.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_107.ctl
e:\tpcd814_work\load\log\lineitem_5_107.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 107 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_108.ctl
e:\tpcd814_work\load\log\lineitem_1_108.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_108.ctl
e:\tpcd814_work\load\log\lineitem_2_108.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_108.ctl
e:\tpcd814_work\load\log\lineitem_3_108.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_108.ctl
e:\tpcd814_work\load\log\lineitem_4_108.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_108.ctl
e:\tpcd814_work\load\log\lineitem_5_108.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 108 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_109.ctl
e:\tpcd814_work\load\log\lineitem_1_109.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_109.ctl
e:\tpcd814_work\load\log\lineitem_2_109.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_109.ctl
e:\tpcd814_work\load\log\lineitem_3_109.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_109.ctl
e:\tpcd814_work\load\log\lineitem_4_109.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_109.ctl
e:\tpcd814_work\load\log\lineitem_5_109.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 109 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_110.ctl
e:\tpcd814_work\load\log\lineitem_1_110.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_110.ctl
```

```
e:\tpcd814_work\load\log\lineitem_2_110.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_110.ctl
e:\tpcd814_work\load\log\lineitem_3_110.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_110.ctl
e:\tpcd814_work\load\log\lineitem_4_110.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_110.ctl
e:\tpcd814_work\load\log\lineitem_5_110.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 110 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_111.ctl
e:\tpcd814_work\load\log\lineitem_1_111.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_111.ctl
e:\tpcd814_work\load\log\lineitem_2_111.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_111.ctl
e:\tpcd814_work\load\log\lineitem_3_111.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_111.ctl
e:\tpcd814_work\load\log\lineitem_4_111.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_111.ctl
e:\tpcd814_work\load\log\lineitem_5_111.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 111 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_112.ctl
e:\tpcd814_work\load\log\lineitem_1_112.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_112.ctl
e:\tpcd814_work\load\log\lineitem_2_112.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_112.ctl
e:\tpcd814_work\load\log\lineitem_3_112.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_112.ctl
e:\tpcd814_work\load\log\lineitem_4_112.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_112.ctl
e:\tpcd814_work\load\log\lineitem_5_112.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 112 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_113.ctl
e:\tpcd814_work\load\log\lineitem_1_113.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_113.ctl
e:\tpcd814_work\load\log\lineitem_2_113.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_113.ctl
e:\tpcd814_work\load\log\lineitem_3_113.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_113.ctl
e:\tpcd814_work\load\log\lineitem_4_113.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_113.ctl
e:\tpcd814_work\load\log\lineitem_5_113.log
direct=true parallel=true
}
```

```
*wait
*time=End of load for Partition: 113 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_114.ctl
e:\tpcd814_work\load\log\lineitem_1_114.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_114.ctl
e:\tpcd814_work\load\log\lineitem_2_114.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_114.ctl
e:\tpcd814_work\load\log\lineitem_3_114.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_114.ctl
e:\tpcd814_work\load\log\lineitem_4_114.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_114.ctl
e:\tpcd814_work\load\log\lineitem_5_114.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 114 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_115.ctl
e:\tpcd814_work\load\log\lineitem_1_115.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_115.ctl
e:\tpcd814_work\load\log\lineitem_2_115.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_115.ctl
e:\tpcd814_work\load\log\lineitem_3_115.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_115.ctl
e:\tpcd814_work\load\log\lineitem_4_115.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_115.ctl
e:\tpcd814_work\load\log\lineitem_5_115.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 115 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_116.ctl
e:\tpcd814_work\load\log\lineitem_1_116.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_116.ctl
e:\tpcd814_work\load\log\lineitem_2_116.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_116.ctl
e:\tpcd814_work\load\log\lineitem_3_116.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_116.ctl
e:\tpcd814_work\load\log\lineitem_4_116.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_116.ctl
e:\tpcd814_work\load\log\lineitem_5_116.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 116 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_117.ctl
e:\tpcd814_work\load\log\lineitem_1_117.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_117.ctl
e:\tpcd814_work\load\log\lineitem_2_117.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_117.ctl
e:\tpcd814_work\load\log\lineitem_3_117.log
direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_117.ctl
e:\tpcd814_work\load\log\lineitem_4_117.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_117.ctl
e:\tpcd814_work\load\log\lineitem_5_117.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 117 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_118.ctl
e:\tpcd814_work\load\log\lineitem_1_118.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_118.ctl
e:\tpcd814_work\load\log\lineitem_2_118.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_118.ctl
e:\tpcd814_work\load\log\lineitem_3_118.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_118.ctl
e:\tpcd814_work\load\log\lineitem_4_118.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_118.ctl
e:\tpcd814_work\load\log\lineitem_5_118.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 118 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_119.ctl
e:\tpcd814_work\load\log\lineitem_1_119.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_119.ctl
e:\tpcd814_work\load\log\lineitem_2_119.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_119.ctl
e:\tpcd814_work\load\log\lineitem_3_119.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_119.ctl
e:\tpcd814_work\load\log\lineitem_4_119.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_119.ctl
e:\tpcd814_work\load\log\lineitem_5_119.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 119 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_120.ctl
e:\tpcd814_work\load\log\lineitem_1_120.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_120.ctl
e:\tpcd814_work\load\log\lineitem_2_120.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_120.ctl
e:\tpcd814_work\load\log\lineitem_3_120.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_120.ctl
e:\tpcd814_work\load\log\lineitem_4_120.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_120.ctl
e:\tpcd814_work\load\log\lineitem_5_120.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 120 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_121.ctl
e:\tpcd814_work\load\log\lineitem_1_121.log
direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_121.ctl
e:\tpcd814_work\load\log\lineitem_2_121.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_121.ctl
e:\tpcd814_work\load\log\lineitem_3_121.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_121.ctl
e:\tpcd814_work\load\log\lineitem_4_121.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_121.ctl
e:\tpcd814_work\load\log\lineitem_5_121.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 121 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_122.ctl
e:\tpcd814_work\load\log\lineitem_1_122.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_122.ctl
e:\tpcd814_work\load\log\lineitem_2_122.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_122.ctl
e:\tpcd814_work\load\log\lineitem_3_122.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_122.ctl
e:\tpcd814_work\load\log\lineitem_4_122.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_122.ctl
e:\tpcd814_work\load\log\lineitem_5_122.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 122 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_123.ctl
e:\tpcd814_work\load\log\lineitem_1_123.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_123.ctl
e:\tpcd814_work\load\log\lineitem_2_123.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_123.ctl
e:\tpcd814_work\load\log\lineitem_3_123.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_123.ctl
e:\tpcd814_work\load\log\lineitem_4_123.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_123.ctl
e:\tpcd814_work\load\log\lineitem_5_123.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 123 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_124.ctl
e:\tpcd814_work\load\log\lineitem_1_124.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_124.ctl
e:\tpcd814_work\load\log\lineitem_2_124.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_124.ctl
e:\tpcd814_work\load\log\lineitem_3_124.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_124.ctl
e:\tpcd814_work\load\log\lineitem_4_124.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\lineitem_5_124.ctl
e:\tpcd814_work\load\log\lineitem_5_124.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 124 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_125.ctl
e:\tpcd814_work\load\log\lineitem_1_125.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_125.ctl
e:\tpcd814_work\load\log\lineitem_2_125.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_125.ctl
e:\tpcd814_work\load\log\lineitem_3_125.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_125.ctl
e:\tpcd814_work\load\log\lineitem_4_125.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_125.ctl
e:\tpcd814_work\load\log\lineitem_5_125.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 125 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_126.ctl
e:\tpcd814_work\load\log\lineitem_1_126.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_126.ctl
e:\tpcd814_work\load\log\lineitem_2_126.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_126.ctl
e:\tpcd814_work\load\log\lineitem_3_126.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_126.ctl
e:\tpcd814_work\load\log\lineitem_4_126.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_126.ctl
e:\tpcd814_work\load\log\lineitem_5_126.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 126 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_127.ctl
e:\tpcd814_work\load\log\lineitem_1_127.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_127.ctl
e:\tpcd814_work\load\log\lineitem_2_127.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_127.ctl
e:\tpcd814_work\load\log\lineitem_3_127.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_127.ctl
e:\tpcd814_work\load\log\lineitem_4_127.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_127.ctl
e:\tpcd814_work\load\log\lineitem_5_127.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 127 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_128.ctl
e:\tpcd814_work\load\log\lineitem_1_128.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_128.ctl
e:\tpcd814_work\load\log\lineitem_2_128.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\lineitem_3_128.ctl
e:\tpcd814_work\load\log\lineitem_3_128.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_128.ctl
e:\tpcd814_work\load\log\lineitem_4_128.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_128.ctl
e:\tpcd814_work\load\log\lineitem_5_128.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 128 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_129.ctl
e:\tpcd814_work\load\log\lineitem_1_129.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_129.ctl
e:\tpcd814_work\load\log\lineitem_2_129.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_129.ctl
e:\tpcd814_work\load\log\lineitem_3_129.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_129.ctl
e:\tpcd814_work\load\log\lineitem_4_129.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_129.ctl
e:\tpcd814_work\load\log\lineitem_5_129.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 129 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_130.ctl
e:\tpcd814_work\load\log\lineitem_1_130.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_130.ctl
e:\tpcd814_work\load\log\lineitem_2_130.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_130.ctl
e:\tpcd814_work\load\log\lineitem_3_130.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_130.ctl
e:\tpcd814_work\load\log\lineitem_4_130.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_130.ctl
e:\tpcd814_work\load\log\lineitem_5_130.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 130 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_131.ctl
e:\tpcd814_work\load\log\lineitem_1_131.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_131.ctl
e:\tpcd814_work\load\log\lineitem_2_131.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_131.ctl
e:\tpcd814_work\load\log\lineitem_3_131.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_131.ctl
e:\tpcd814_work\load\log\lineitem_4_131.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_131.ctl
e:\tpcd814_work\load\log\lineitem_5_131.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 131 for Table:
lineitem
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\lineitem_1_132.ctl
e:\tpcd814_work\load\log\lineitem_1_132.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_132.ctl
e:\tpcd814_work\load\log\lineitem_2_132.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_132.ctl
e:\tpcd814_work\load\log\lineitem_3_132.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_132.ctl
e:\tpcd814_work\load\log\lineitem_4_132.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_132.ctl
e:\tpcd814_work\load\log\lineitem_5_132.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 132 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_133.ctl
e:\tpcd814_work\load\log\lineitem_1_133.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_133.ctl
e:\tpcd814_work\load\log\lineitem_2_133.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_133.ctl
e:\tpcd814_work\load\log\lineitem_3_133.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_133.ctl
e:\tpcd814_work\load\log\lineitem_4_133.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_133.ctl
e:\tpcd814_work\load\log\lineitem_5_133.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 133 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_134.ctl
e:\tpcd814_work\load\log\lineitem_1_134.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_134.ctl
e:\tpcd814_work\load\log\lineitem_2_134.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_134.ctl
e:\tpcd814_work\load\log\lineitem_3_134.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_134.ctl
e:\tpcd814_work\load\log\lineitem_4_134.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_134.ctl
e:\tpcd814_work\load\log\lineitem_5_134.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 134 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_135.ctl
e:\tpcd814_work\load\log\lineitem_1_135.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_135.ctl
e:\tpcd814_work\load\log\lineitem_2_135.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_135.ctl
e:\tpcd814_work\load\log\lineitem_3_135.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_135.ctl
e:\tpcd814_work\load\log\lineitem_4_135.log
direct=true parallel=true
```

```
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_135.ctl
e:\tpcd814_work\load\log\lineitem_5_135.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 135 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_136.ctl
e:\tpcd814_work\load\log\lineitem_1_136.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_136.ctl
e:\tpcd814_work\load\log\lineitem_2_136.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_136.ctl
e:\tpcd814_work\load\log\lineitem_3_136.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_136.ctl
e:\tpcd814_work\load\log\lineitem_4_136.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_136.ctl
e:\tpcd814_work\load\log\lineitem_5_136.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 136 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_137.ctl
e:\tpcd814_work\load\log\lineitem_1_137.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_137.ctl
e:\tpcd814_work\load\log\lineitem_2_137.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_137.ctl
e:\tpcd814_work\load\log\lineitem_3_137.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_137.ctl
e:\tpcd814_work\load\log\lineitem_4_137.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_137.ctl
e:\tpcd814_work\load\log\lineitem_5_137.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 137 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_138.ctl
e:\tpcd814_work\load\log\lineitem_1_138.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_138.ctl
e:\tpcd814_work\load\log\lineitem_2_138.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_138.ctl
e:\tpcd814_work\load\log\lineitem_3_138.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_138.ctl
e:\tpcd814_work\load\log\lineitem_4_138.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_138.ctl
e:\tpcd814_work\load\log\lineitem_5_138.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 138 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_139.ctl
e:\tpcd814_work\load\log\lineitem_1_139.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_139.ctl
e:\tpcd814_work\load\log\lineitem_2_139.log
direct=true parallel=true
```

```
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_139.ctl
e:\tpcd814_work\load\log\lineitem_3_139.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_139.ctl
e:\tpcd814_work\load\log\lineitem_4_139.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_139.ctl
e:\tpcd814_work\load\log\lineitem_5_139.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 139 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_140.ctl
e:\tpcd814_work\load\log\lineitem_1_140.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_140.ctl
e:\tpcd814_work\load\log\lineitem_2_140.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_140.ctl
e:\tpcd814_work\load\log\lineitem_3_140.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_140.ctl
e:\tpcd814_work\load\log\lineitem_4_140.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_140.ctl
e:\tpcd814_work\load\log\lineitem_5_140.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 140 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_141.ctl
e:\tpcd814_work\load\log\lineitem_1_141.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_141.ctl
e:\tpcd814_work\load\log\lineitem_2_141.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_141.ctl
e:\tpcd814_work\load\log\lineitem_3_141.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_141.ctl
e:\tpcd814_work\load\log\lineitem_4_141.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_141.ctl
e:\tpcd814_work\load\log\lineitem_5_141.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 141 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_142.ctl
e:\tpcd814_work\load\log\lineitem_1_142.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_142.ctl
e:\tpcd814_work\load\log\lineitem_2_142.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_142.ctl
e:\tpcd814_work\load\log\lineitem_3_142.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_142.ctl
e:\tpcd814_work\load\log\lineitem_4_142.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_142.ctl
e:\tpcd814_work\load\log\lineitem_5_142.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 142 for Table:
```

```
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_143.ctl
e:\tpcd814_work\load\log\lineitem_1_143.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_143.ctl
e:\tpcd814_work\load\log\lineitem_2_143.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_143.ctl
e:\tpcd814_work\load\log\lineitem_3_143.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_143.ctl
e:\tpcd814_work\load\log\lineitem_4_143.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_143.ctl
e:\tpcd814_work\load\log\lineitem_5_143.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 143 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_144.ctl
e:\tpcd814_work\load\log\lineitem_1_144.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_144.ctl
e:\tpcd814_work\load\log\lineitem_2_144.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_144.ctl
e:\tpcd814_work\load\log\lineitem_3_144.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_144.ctl
e:\tpcd814_work\load\log\lineitem_4_144.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_144.ctl
e:\tpcd814_work\load\log\lineitem_5_144.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 144 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_145.ctl
e:\tpcd814_work\load\log\lineitem_1_145.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_145.ctl
e:\tpcd814_work\load\log\lineitem_2_145.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_145.ctl
e:\tpcd814_work\load\log\lineitem_3_145.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_145.ctl
e:\tpcd814_work\load\log\lineitem_4_145.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_145.ctl
e:\tpcd814_work\load\log\lineitem_5_145.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 145 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_146.ctl
e:\tpcd814_work\load\log\lineitem_1_146.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_146.ctl
e:\tpcd814_work\load\log\lineitem_2_146.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_146.ctl
e:\tpcd814_work\load\log\lineitem_3_146.log
direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_146.ctl
e:\tpcd814_work\load\log\lineitem_4_146.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_146.ctl
e:\tpcd814_work\load\log\lineitem_5_146.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 146 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_147.ctl
e:\tpcd814_work\load\log\lineitem_1_147.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_147.ctl
e:\tpcd814_work\load\log\lineitem_2_147.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_147.ctl
e:\tpcd814_work\load\log\lineitem_3_147.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_147.ctl
e:\tpcd814_work\load\log\lineitem_4_147.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_147.ctl
e:\tpcd814_work\load\log\lineitem_5_147.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 147 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_148.ctl
e:\tpcd814_work\load\log\lineitem_1_148.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_148.ctl
e:\tpcd814_work\load\log\lineitem_2_148.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_148.ctl
e:\tpcd814_work\load\log\lineitem_3_148.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_148.ctl
e:\tpcd814_work\load\log\lineitem_4_148.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_148.ctl
e:\tpcd814_work\load\log\lineitem_5_148.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 148 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_149.ctl
e:\tpcd814_work\load\log\lineitem_1_149.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_149.ctl
e:\tpcd814_work\load\log\lineitem_2_149.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_149.ctl
e:\tpcd814_work\load\log\lineitem_3_149.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_149.ctl
e:\tpcd814_work\load\log\lineitem_4_149.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_149.ctl
e:\tpcd814_work\load\log\lineitem_5_149.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 149 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_150.ctl
e:\tpcd814_work\load\log\lineitem_1_150.log
direct=true parallel=true
}
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_150.ctl
e:\tpcd814_work\load\log\lineitem_2_150.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_150.ctl
e:\tpcd814_work\load\log\lineitem_3_150.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_150.ctl
e:\tpcd814_work\load\log\lineitem_4_150.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_150.ctl
e:\tpcd814_work\load\log\lineitem_5_150.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 150 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_151.ctl
e:\tpcd814_work\load\log\lineitem_1_151.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_151.ctl
e:\tpcd814_work\load\log\lineitem_2_151.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_151.ctl
e:\tpcd814_work\load\log\lineitem_3_151.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_151.ctl
e:\tpcd814_work\load\log\lineitem_4_151.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_151.ctl
e:\tpcd814_work\load\log\lineitem_5_151.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 151 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_152.ctl
e:\tpcd814_work\load\log\lineitem_1_152.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_152.ctl
e:\tpcd814_work\load\log\lineitem_2_152.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_152.ctl
e:\tpcd814_work\load\log\lineitem_3_152.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_152.ctl
e:\tpcd814_work\load\log\lineitem_4_152.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_152.ctl
e:\tpcd814_work\load\log\lineitem_5_152.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 152 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_153.ctl
e:\tpcd814_work\load\log\lineitem_1_153.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_153.ctl
e:\tpcd814_work\load\log\lineitem_2_153.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_153.ctl
e:\tpcd814_work\load\log\lineitem_3_153.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_153.ctl
e:\tpcd814_work\load\log\lineitem_4_153.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_153.ctl
e:\tpcd814_work\load\log\lineitem_5_153.log
```

```
direct=true parallel=true
}
*wait
*time=End of load for Partition: 153 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_154.ctl
e:\tpcd814_work\load\log\lineitem_1_154.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_154.ctl
e:\tpcd814_work\load\log\lineitem_2_154.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_154.ctl
e:\tpcd814_work\load\log\lineitem_3_154.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_154.ctl
e:\tpcd814_work\load\log\lineitem_4_154.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_154.ctl
e:\tpcd814_work\load\log\lineitem_5_154.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 154 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_155.ctl
e:\tpcd814_work\load\log\lineitem_1_155.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_155.ctl
e:\tpcd814_work\load\log\lineitem_2_155.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_155.ctl
e:\tpcd814_work\load\log\lineitem_3_155.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_155.ctl
e:\tpcd814_work\load\log\lineitem_4_155.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_155.ctl
e:\tpcd814_work\load\log\lineitem_5_155.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 155 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_156.ctl
e:\tpcd814_work\load\log\lineitem_1_156.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_156.ctl
e:\tpcd814_work\load\log\lineitem_2_156.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_156.ctl
e:\tpcd814_work\load\log\lineitem_3_156.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_156.ctl
e:\tpcd814_work\load\log\lineitem_4_156.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_156.ctl
e:\tpcd814_work\load\log\lineitem_5_156.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 156 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_157.ctl
e:\tpcd814_work\load\log\lineitem_1_157.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_157.ctl
e:\tpcd814_work\load\log\lineitem_2_157.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_157.ctl
e:\tpcd814_work\load\log\lineitem_3_157.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_157.ctl
e:\tpcd814_work\load\log\lineitem_4_157.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_157.ctl
e:\tpcd814_work\load\log\lineitem_5_157.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 157 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_158.ctl
e:\tpcd814_work\load\log\lineitem_1_158.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_158.ctl
e:\tpcd814_work\load\log\lineitem_2_158.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_158.ctl
e:\tpcd814_work\load\log\lineitem_3_158.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_158.ctl
e:\tpcd814_work\load\log\lineitem_4_158.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_158.ctl
e:\tpcd814_work\load\log\lineitem_5_158.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 158 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_159.ctl
e:\tpcd814_work\load\log\lineitem_1_159.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_159.ctl
e:\tpcd814_work\load\log\lineitem_2_159.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_159.ctl
e:\tpcd814_work\load\log\lineitem_3_159.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_159.ctl
e:\tpcd814_work\load\log\lineitem_4_159.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_159.ctl
e:\tpcd814_work\load\log\lineitem_5_159.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 159 for Table:
lineitem
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_1_160.ctl
e:\tpcd814_work\load\log\lineitem_1_160.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_2_160.ctl
e:\tpcd814_work\load\log\lineitem_2_160.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_3_160.ctl
e:\tpcd814_work\load\log\lineitem_3_160.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_4_160.ctl
e:\tpcd814_work\load\log\lineitem_4_160.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\lineitem_5_160.ctl
e:\tpcd814_work\load\log\lineitem_5_160.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 160 for Table:
lineitem
*wait
*time=End lineitem load
*time=Begin orders load
*load
{
```

```
tpcd/tpcd
e:\tpcd814_work\load\control\orders_1.ctl
e:\tpcd814_work\load\log\orders_1.log direct=true
parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\orders_2.ctl
e:\tpcd814_work\load\log\orders_2.log direct=true
parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\orders_3.ctl
e:\tpcd814_work\load\log\orders_3.log direct=true
parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\orders_4.ctl
e:\tpcd814_work\load\log\orders_4.log direct=true
parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\orders_5.ctl
e:\tpcd814_work\load\log\orders_5.log direct=true
parallel=true
}
*wait
*time=End orders load
*time=Begin parts load
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_1.ctl
e:\tpcd814_work\load\log\parts_1_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_1.ctl
e:\tpcd814_work\load\log\parts_2_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_1.ctl
e:\tpcd814_work\load\log\parts_3_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_1.ctl
e:\tpcd814_work\load\log\parts_4_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_1.ctl
e:\tpcd814_work\load\log\parts_5_1.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 1 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_2.ctl
e:\tpcd814_work\load\log\parts_1_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_2.ctl
e:\tpcd814_work\load\log\parts_2_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_2.ctl
e:\tpcd814_work\load\log\parts_3_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_2.ctl
e:\tpcd814_work\load\log\parts_4_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_2.ctl
e:\tpcd814_work\load\log\parts_5_2.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 2 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_3.ctl
e:\tpcd814_work\load\log\parts_1_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_3.ctl
e:\tpcd814_work\load\log\parts_2_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_3.ctl
e:\tpcd814_work\load\log\parts_3_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_3.ctl
e:\tpcd814_work\load\log\parts_4_3.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_3.ctl
e:\tpcd814_work\load\log\parts_5_3.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 3 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_4.ctl
e:\tpcd814_work\load\log\parts_1_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_4.ctl
e:\tpcd814_work\load\log\parts_2_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_4.ctl
e:\tpcd814_work\load\log\parts_3_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_4.ctl
e:\tpcd814_work\load\log\parts_4_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_4.ctl
e:\tpcd814_work\load\log\parts_5_4.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 4 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_5.ctl
e:\tpcd814_work\load\log\parts_1_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_5.ctl
e:\tpcd814_work\load\log\parts_2_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_5.ctl
e:\tpcd814_work\load\log\parts_3_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_5.ctl
e:\tpcd814_work\load\log\parts_4_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_5.ctl
e:\tpcd814_work\load\log\parts_5_5.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 5 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_6.ctl
e:\tpcd814_work\load\log\parts_1_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_6.ctl
e:\tpcd814_work\load\log\parts_2_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_6.ctl
e:\tpcd814_work\load\log\parts_3_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_6.ctl
e:\tpcd814_work\load\log\parts_4_6.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_6.ctl
e:\tpcd814_work\load\log\parts_5_6.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 6 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_7.ctl
e:\tpcd814_work\load\log\parts_1_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_7.ctl
e:\tpcd814_work\load\log\parts_2_7.log
```

```
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_7.ctl
e:\tpcd814_work\load\log\parts_3_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_7.ctl
e:\tpcd814_work\load\log\parts_4_7.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_7.ctl
e:\tpcd814_work\load\log\parts_5_7.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 7 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_8.ctl
e:\tpcd814_work\load\log\parts_1_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_8.ctl
e:\tpcd814_work\load\log\parts_2_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_8.ctl
e:\tpcd814_work\load\log\parts_3_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_8.ctl
e:\tpcd814_work\load\log\parts_4_8.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_8.ctl
e:\tpcd814_work\load\log\parts_5_8.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 8 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_9.ctl
e:\tpcd814_work\load\log\parts_1_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_9.ctl
e:\tpcd814_work\load\log\parts_2_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_9.ctl
e:\tpcd814_work\load\log\parts_3_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_9.ctl
e:\tpcd814_work\load\log\parts_4_9.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_9.ctl
e:\tpcd814_work\load\log\parts_5_9.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 9 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_10.ctl
e:\tpcd814_work\load\log\parts_1_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_10.ctl
e:\tpcd814_work\load\log\parts_2_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_10.ctl
e:\tpcd814_work\load\log\parts_3_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_10.ctl
e:\tpcd814_work\load\log\parts_4_10.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_10.ctl
e:\tpcd814_work\load\log\parts_5_10.log
direct=true parallel=true
}
*wait
```

```
*time=End of load for Partition: 10 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_11.ctl
e:\tpcd814_work\load\log\parts_1_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_11.ctl
e:\tpcd814_work\load\log\parts_2_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_11.ctl
e:\tpcd814_work\load\log\parts_3_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_11.ctl
e:\tpcd814_work\load\log\parts_4_11.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_11.ctl
e:\tpcd814_work\load\log\parts_5_11.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 11 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_12.ctl
e:\tpcd814_work\load\log\parts_1_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_12.ctl
e:\tpcd814_work\load\log\parts_2_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_12.ctl
e:\tpcd814_work\load\log\parts_3_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_12.ctl
e:\tpcd814_work\load\log\parts_4_12.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_12.ctl
e:\tpcd814_work\load\log\parts_5_12.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 12 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_13.ctl
e:\tpcd814_work\load\log\parts_1_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_13.ctl
e:\tpcd814_work\load\log\parts_2_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_13.ctl
e:\tpcd814_work\load\log\parts_3_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_13.ctl
e:\tpcd814_work\load\log\parts_4_13.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_13.ctl
e:\tpcd814_work\load\log\parts_5_13.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 13 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_14.ctl
e:\tpcd814_work\load\log\parts_1_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_14.ctl
e:\tpcd814_work\load\log\parts_2_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_14.ctl
e:\tpcd814_work\load\log\parts_3_14.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_14.ctl
e:\tpcd814_work\load\log\parts_4_14.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_14.ctl
e:\tpcd814_work\load\log\parts_5_14.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 14 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_15.ctl
e:\tpcd814_work\load\log\parts_1_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_15.ctl
e:\tpcd814_work\load\log\parts_2_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_15.ctl
e:\tpcd814_work\load\log\parts_3_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_15.ctl
e:\tpcd814_work\load\log\parts_4_15.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_15.ctl
e:\tpcd814_work\load\log\parts_5_15.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 15 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_16.ctl
e:\tpcd814_work\load\log\parts_1_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_16.ctl
e:\tpcd814_work\load\log\parts_2_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_16.ctl
e:\tpcd814_work\load\log\parts_3_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_16.ctl
e:\tpcd814_work\load\log\parts_4_16.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_16.ctl
e:\tpcd814_work\load\log\parts_5_16.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 16 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_17.ctl
e:\tpcd814_work\load\log\parts_1_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_17.ctl
e:\tpcd814_work\load\log\parts_2_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_17.ctl
e:\tpcd814_work\load\log\parts_3_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_17.ctl
e:\tpcd814_work\load\log\parts_4_17.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_17.ctl
e:\tpcd814_work\load\log\parts_5_17.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 17 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_18.ctl
e:\tpcd814_work\load\log\parts_1_18.log
direct=true parallel=true
}
*load
```

```
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_18.ctl
e:\tpcd814_work\load\log\parts_2_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_18.ctl
e:\tpcd814_work\load\log\parts_3_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_18.ctl
e:\tpcd814_work\load\log\parts_4_18.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_18.ctl
e:\tpcd814_work\load\log\parts_5_18.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 18 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_19.ctl
e:\tpcd814_work\load\log\parts_1_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_19.ctl
e:\tpcd814_work\load\log\parts_2_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_19.ctl
e:\tpcd814_work\load\log\parts_3_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_19.ctl
e:\tpcd814_work\load\log\parts_4_19.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_19.ctl
e:\tpcd814_work\load\log\parts_5_19.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 19 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_20.ctl
e:\tpcd814_work\load\log\parts_1_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_20.ctl
e:\tpcd814_work\load\log\parts_2_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_20.ctl
e:\tpcd814_work\load\log\parts_3_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_20.ctl
e:\tpcd814_work\load\log\parts_4_20.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_20.ctl
e:\tpcd814_work\load\log\parts_5_20.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 20 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_21.ctl
e:\tpcd814_work\load\log\parts_1_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_21.ctl
e:\tpcd814_work\load\log\parts_2_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_21.ctl
e:\tpcd814_work\load\log\parts_3_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_21.ctl
e:\tpcd814_work\load\log\parts_4_21.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_21.ctl
```

```
e:\tpcd814_work\load\log\parts_5_21.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 21 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_22.ctl
e:\tpcd814_work\load\log\parts_1_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_22.ctl
e:\tpcd814_work\load\log\parts_2_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_22.ctl
e:\tpcd814_work\load\log\parts_3_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_22.ctl
e:\tpcd814_work\load\log\parts_4_22.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_22.ctl
e:\tpcd814_work\load\log\parts_5_22.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 22 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_23.ctl
e:\tpcd814_work\load\log\parts_1_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_23.ctl
e:\tpcd814_work\load\log\parts_2_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_23.ctl
e:\tpcd814_work\load\log\parts_3_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_23.ctl
e:\tpcd814_work\load\log\parts_4_23.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_23.ctl
e:\tpcd814_work\load\log\parts_5_23.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 23 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_24.ctl
e:\tpcd814_work\load\log\parts_1_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_24.ctl
e:\tpcd814_work\load\log\parts_2_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_24.ctl
e:\tpcd814_work\load\log\parts_3_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_24.ctl
e:\tpcd814_work\load\log\parts_4_24.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_24.ctl
e:\tpcd814_work\load\log\parts_5_24.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 24 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_25.ctl
e:\tpcd814_work\load\log\parts_1_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_25.ctl
e:\tpcd814_work\load\log\parts_2_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_25.ctl
```

```
e:\tpcd814_work\load\log\parts_3_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_25.ctl
e:\tpcd814_work\load\log\parts_4_25.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_25.ctl
e:\tpcd814_work\load\log\parts_5_25.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 25 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_26.ctl
e:\tpcd814_work\load\log\parts_1_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_26.ctl
e:\tpcd814_work\load\log\parts_2_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_26.ctl
e:\tpcd814_work\load\log\parts_3_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_26.ctl
e:\tpcd814_work\load\log\parts_4_26.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_26.ctl
e:\tpcd814_work\load\log\parts_5_26.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 26 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_27.ctl
e:\tpcd814_work\load\log\parts_1_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_27.ctl
e:\tpcd814_work\load\log\parts_2_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_27.ctl
e:\tpcd814_work\load\log\parts_3_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_27.ctl
e:\tpcd814_work\load\log\parts_4_27.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_27.ctl
e:\tpcd814_work\load\log\parts_5_27.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 27 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_28.ctl
e:\tpcd814_work\load\log\parts_1_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_28.ctl
e:\tpcd814_work\load\log\parts_2_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_28.ctl
e:\tpcd814_work\load\log\parts_3_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_28.ctl
e:\tpcd814_work\load\log\parts_4_28.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_28.ctl
e:\tpcd814_work\load\log\parts_5_28.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 28 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_29.ctl
```

```
e:\tpcd814_work\load\log\parts_1_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_29.ctl
e:\tpcd814_work\load\log\parts_2_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_29.ctl
e:\tpcd814_work\load\log\parts_3_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_29.ctl
e:\tpcd814_work\load\log\parts_4_29.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_29.ctl
e:\tpcd814_work\load\log\parts_5_29.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 29 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_30.ctl
e:\tpcd814_work\load\log\parts_1_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_30.ctl
e:\tpcd814_work\load\log\parts_2_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_30.ctl
e:\tpcd814_work\load\log\parts_3_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_30.ctl
e:\tpcd814_work\load\log\parts_4_30.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_30.ctl
e:\tpcd814_work\load\log\parts_5_30.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 30 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_31.ctl
e:\tpcd814_work\load\log\parts_1_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_31.ctl
e:\tpcd814_work\load\log\parts_2_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_31.ctl
e:\tpcd814_work\load\log\parts_3_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_31.ctl
e:\tpcd814_work\load\log\parts_4_31.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_31.ctl
e:\tpcd814_work\load\log\parts_5_31.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 31 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_32.ctl
e:\tpcd814_work\load\log\parts_1_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_32.ctl
e:\tpcd814_work\load\log\parts_2_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_32.ctl
e:\tpcd814_work\load\log\parts_3_32.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_32.ctl
e:\tpcd814_work\load\log\parts_4_32.log
direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_32.ctl
e:\tpcd814_work\load\log\parts_5_32.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 32 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_33.ctl
e:\tpcd814_work\load\log\parts_1_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_33.ctl
e:\tpcd814_work\load\log\parts_2_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_33.ctl
e:\tpcd814_work\load\log\parts_3_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_33.ctl
e:\tpcd814_work\load\log\parts_4_33.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_33.ctl
e:\tpcd814_work\load\log\parts_5_33.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 33 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_34.ctl
e:\tpcd814_work\load\log\parts_1_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_34.ctl
e:\tpcd814_work\load\log\parts_2_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_34.ctl
e:\tpcd814_work\load\log\parts_3_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_34.ctl
e:\tpcd814_work\load\log\parts_4_34.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_34.ctl
e:\tpcd814_work\load\log\parts_5_34.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 34 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_35.ctl
e:\tpcd814_work\load\log\parts_1_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_35.ctl
e:\tpcd814_work\load\log\parts_2_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_35.ctl
e:\tpcd814_work\load\log\parts_3_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_35.ctl
e:\tpcd814_work\load\log\parts_4_35.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_35.ctl
e:\tpcd814_work\load\log\parts_5_35.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 35 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_36.ctl
e:\tpcd814_work\load\log\parts_1_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_36.ctl
e:\tpcd814_work\load\log\parts_2_36.log
direct=true parallel=true
}
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_36.ctl
e:\tpcd814_work\load\log\parts_3_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_36.ctl
e:\tpcd814_work\load\log\parts_4_36.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_36.ctl
e:\tpcd814_work\load\log\parts_5_36.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 36 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_37.ctl
e:\tpcd814_work\load\log\parts_1_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_37.ctl
e:\tpcd814_work\load\log\parts_2_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_37.ctl
e:\tpcd814_work\load\log\parts_3_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_37.ctl
e:\tpcd814_work\load\log\parts_4_37.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_37.ctl
e:\tpcd814_work\load\log\parts_5_37.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 37 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_38.ctl
e:\tpcd814_work\load\log\parts_1_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_38.ctl
e:\tpcd814_work\load\log\parts_2_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_38.ctl
e:\tpcd814_work\load\log\parts_3_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_38.ctl
e:\tpcd814_work\load\log\parts_4_38.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_38.ctl
e:\tpcd814_work\load\log\parts_5_38.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 38 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_39.ctl
e:\tpcd814_work\load\log\parts_1_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_39.ctl
e:\tpcd814_work\load\log\parts_2_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_39.ctl
e:\tpcd814_work\load\log\parts_3_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_39.ctl
e:\tpcd814_work\load\log\parts_4_39.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_39.ctl
e:\tpcd814_work\load\log\parts_5_39.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 39 for Table:
parts
```

```
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_40.ctl
e:\tpcd814_work\load\log\parts_1_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_40.ctl
e:\tpcd814_work\load\log\parts_2_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_40.ctl
e:\tpcd814_work\load\log\parts_3_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_40.ctl
e:\tpcd814_work\load\log\parts_4_40.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_40.ctl
e:\tpcd814_work\load\log\parts_5_40.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 40 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_41.ctl
e:\tpcd814_work\load\log\parts_1_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_41.ctl
e:\tpcd814_work\load\log\parts_2_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_41.ctl
e:\tpcd814_work\load\log\parts_3_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_41.ctl
e:\tpcd814_work\load\log\parts_4_41.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_41.ctl
e:\tpcd814_work\load\log\parts_5_41.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 41 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_42.ctl
e:\tpcd814_work\load\log\parts_1_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_42.ctl
e:\tpcd814_work\load\log\parts_2_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_42.ctl
e:\tpcd814_work\load\log\parts_3_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_42.ctl
e:\tpcd814_work\load\log\parts_4_42.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_42.ctl
e:\tpcd814_work\load\log\parts_5_42.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 42 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_43.ctl
e:\tpcd814_work\load\log\parts_1_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_43.ctl
e:\tpcd814_work\load\log\parts_2_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_43.ctl
e:\tpcd814_work\load\log\parts_3_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\parts_4_43.ctl
e:\tpcd814_work\load\log\parts_4_43.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_43.ctl
e:\tpcd814_work\load\log\parts_5_43.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 43 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_44.ctl
e:\tpcd814_work\load\log\parts_1_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_44.ctl
e:\tpcd814_work\load\log\parts_2_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_44.ctl
e:\tpcd814_work\load\log\parts_3_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_44.ctl
e:\tpcd814_work\load\log\parts_4_44.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_44.ctl
e:\tpcd814_work\load\log\parts_5_44.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 44 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_45.ctl
e:\tpcd814_work\load\log\parts_1_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_45.ctl
e:\tpcd814_work\load\log\parts_2_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_45.ctl
e:\tpcd814_work\load\log\parts_3_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_45.ctl
e:\tpcd814_work\load\log\parts_4_45.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_45.ctl
e:\tpcd814_work\load\log\parts_5_45.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 45 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_46.ctl
e:\tpcd814_work\load\log\parts_1_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_46.ctl
e:\tpcd814_work\load\log\parts_2_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_46.ctl
e:\tpcd814_work\load\log\parts_3_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_46.ctl
e:\tpcd814_work\load\log\parts_4_46.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_46.ctl
e:\tpcd814_work\load\log\parts_5_46.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 46 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_47.ctl
e:\tpcd814_work\load\log\parts_1_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\parts_2_47.ctl
e:\tpcd814_work\load\log\parts_2_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_47.ctl
e:\tpcd814_work\load\log\parts_3_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_47.ctl
e:\tpcd814_work\load\log\parts_4_47.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_47.ctl
e:\tpcd814_work\load\log\parts_5_47.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 47 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_48.ctl
e:\tpcd814_work\load\log\parts_1_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_48.ctl
e:\tpcd814_work\load\log\parts_2_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_48.ctl
e:\tpcd814_work\load\log\parts_3_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_48.ctl
e:\tpcd814_work\load\log\parts_4_48.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_48.ctl
e:\tpcd814_work\load\log\parts_5_48.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 48 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_49.ctl
e:\tpcd814_work\load\log\parts_1_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_49.ctl
e:\tpcd814_work\load\log\parts_2_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_49.ctl
e:\tpcd814_work\load\log\parts_3_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_49.ctl
e:\tpcd814_work\load\log\parts_4_49.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_49.ctl
e:\tpcd814_work\load\log\parts_5_49.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 49 for Table:
parts
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_1_50.ctl
e:\tpcd814_work\load\log\parts_1_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_2_50.ctl
e:\tpcd814_work\load\log\parts_2_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_3_50.ctl
e:\tpcd814_work\load\log\parts_3_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_4_50.ctl
e:\tpcd814_work\load\log\parts_4_50.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\parts_5_50.ctl
e:\tpcd814_work\load\log\parts_5_50.log
direct=true parallel=true
```

```
}
*wait
*time=End of load for Partition: 50 for Table:
parts
*wait
*time=End parts load
*time=Begin customer load
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_1_1.ctl
e:\tpcd814_work\load\log\customer_1_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_2_1.ctl
e:\tpcd814_work\load\log\customer_2_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_3_1.ctl
e:\tpcd814_work\load\log\customer_3_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_4_1.ctl
e:\tpcd814_work\load\log\customer_4_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_5_1.ctl
e:\tpcd814_work\load\log\customer_5_1.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 1 for Table:
customer
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_1_2.ctl
e:\tpcd814_work\load\log\customer_1_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_2_2.ctl
e:\tpcd814_work\load\log\customer_2_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_3_2.ctl
e:\tpcd814_work\load\log\customer_3_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_4_2.ctl
e:\tpcd814_work\load\log\customer_4_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_5_2.ctl
e:\tpcd814_work\load\log\customer_5_2.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 2 for Table:
customer
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_1_3.ctl
e:\tpcd814_work\load\log\customer_1_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_2_3.ctl
e:\tpcd814_work\load\log\customer_2_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_3_3.ctl
e:\tpcd814_work\load\log\customer_3_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_4_3.ctl
e:\tpcd814_work\load\log\customer_4_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_5_3.ctl
e:\tpcd814_work\load\log\customer_5_3.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 3 for Table:
customer
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_1_4.ctl
e:\tpcd814_work\load\log\customer_1_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_2_4.ctl
e:\tpcd814_work\load\log\customer_2_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
```

```
e:\tpcd814_work\load\control\customer_3_4.ctl
e:\tpcd814_work\load\log\customer_3_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_4_4.ctl
e:\tpcd814_work\load\log\customer_4_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_5_4.ctl
e:\tpcd814_work\load\log\customer_5_4.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 4 for Table:
customer
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_1_5.ctl
e:\tpcd814_work\load\log\customer_1_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_2_5.ctl
e:\tpcd814_work\load\log\customer_2_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_3_5.ctl
e:\tpcd814_work\load\log\customer_3_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_4_5.ctl
e:\tpcd814_work\load\log\customer_4_5.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\customer_5_5.ctl
e:\tpcd814_work\load\log\customer_5_5.log
direct=true parallel=true
}
*wait
*time=End of load for Partition: 5 for Table:
customer
*wait
*time=End customer load
*time=Begin partsupp load
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\partsupp_1.ctl
e:\tpcd814_work\load\log\partsupp_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\partsupp_2.ctl
e:\tpcd814_work\load\log\partsupp_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\partsupp_3.ctl
e:\tpcd814_work\load\log\partsupp_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\partsupp_4.ctl
e:\tpcd814_work\load\log\partsupp_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\partsupp_5.ctl
e:\tpcd814_work\load\log\partsupp_5.log
direct=true parallel=true
}
*wait
*time=End partsupp load
*time=Begin supplier load
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\supplier_1.ctl
e:\tpcd814_work\load\log\supplier_1.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\supplier_2.ctl
e:\tpcd814_work\load\log\supplier_2.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\supplier_3.ctl
e:\tpcd814_work\load\log\supplier_3.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\supplier_4.ctl
e:\tpcd814_work\load\log\supplier_4.log
direct=true parallel=true
}
*load
{
tpcd/tpcd
e:\tpcd814_work\load\control\supplier_5.ctl
e:\tpcd814_work\load\log\supplier_5.log
direct=true parallel=true
}
*wait
*time=End supplier load
*time=Begin nation load
```

```
*load
{
tpcd/tpcd e:\tpcd814_work\load\control\nation.ctl
e:\tpcd814_work\load\log\nation.log direct=true
}
*wait
*wait
*time=End nation load
*time=Begin region load
*load
{
tpcd/tpcd e:\tpcd814_work\load\control\region.ctl
e:\tpcd814_work\load\log\region.log direct=true
}
*wait
*wait
*time=End region load
*wait
*time=Done data population
*wait
*bgoff
%e-dapop
```

100g_816_dbcre.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=1
################################################
############################
# Database Creation Phase
*time=Begin database creation
*sql
{
shutdown normal;
}
*wait
# creating database and initial rollback segment
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora nomount;
create database
  controlfile reuse
  logfile ( '\\.\log_100g1') size 3200m reuse,
          ( '\\.\log_100g2') size 3200m reuse
  datafile '\\.\sys_100g1' size 20m reuse
  maxdatafiles 10000
;
alter tablespace system
  add datafile '\\.\sys_100g2' size 20m reuse;
alter tablespace system
  add datafile '\\.\sys_100g3' size 20m reuse;
alter tablespace system
  add datafile '\\.\sys_100g4' size 20m reuse;
alter tablespace system
  add datafile '\\.\sys_100g5' size 20m reuse;

create public rollback segment t_rs1   storage
(initial 10m next 10m optimal 50m maxextents
unlimited);

alter rollback segment t_rs1 online;

shutdown
}
*wait
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
}
*wait
*sql
{
create tablespace ts_undo
  datafile '\\.\undo_100g1' size 1000m reuse;
}
*wait
# adding tpcd's ts_undo datafiles
*sql
{
alter tablespace ts_undo
  add datafile '\\.\undo_100g2' size 1000m reuse;
alter tablespace ts_undo
  add datafile '\\.\undo_100g3' size 1000m reuse;
alter tablespace ts_undo
  add datafile '\\.\undo_100g4' size 1000m reuse;
alter tablespace ts_undo
  add datafile '\\.\undo_100g5' size 1000m reuse;
}
*wait
# creating extra rollback segments
*sql
{
create public rollback segment r1 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r2 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r3 storage
(initial 10m next 10m optimal 50m maxextents
```

```
unlimited) tablespace ts_undo;
create public rollback segment r4 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r5 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r6 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r7 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r8 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r9 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r10 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r11 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r12 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r13 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r14 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r15 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r16 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r17 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r18 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r19 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r20 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r21 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r22 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r23 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r24 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r25 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r26 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r27 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r28 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r29 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r30 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r31 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r32 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r33 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r34 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r35 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r36 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r37 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r38 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r39 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r40 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r41 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r42 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r43 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r44 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r45 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r46 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r47 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r48 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r49 storage
(initial 10m next 10m optimal 50m maxextents
```

```
unlimited) tablespace ts_undo;
create public rollback segment r50 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r51 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r52 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r53 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r54 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r55 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r56 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r57 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r58 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r59 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r60 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r61 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r62 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r63 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r64 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r65 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r66 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r67 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r68 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r69 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r70 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r71 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r72 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r73 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r74 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r75 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r76 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r77 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r78 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r79 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r80 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r81 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r82 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r83 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r84 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r85 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r86 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r87 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r88 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r89 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r90 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r91 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r92 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r93 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r94 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r95 storage
(initial 10m next 10m optimal 50m maxextents
```

```
unlimited) tablespace ts_undo;
create public rollback segment r96 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r97 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r98 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r99 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r100 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r101 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r102 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r103 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r104 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r105 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r106 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r107 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r108 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r109 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r110 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r111 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r112 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r113 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r114 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r115 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r116 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r117 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r118 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r119 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r120 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r121 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r122 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r123 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r124 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r125 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r126 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r127 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r128 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r129 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r130 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r131 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r132 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r133 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r134 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r135 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r136 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r137 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r138 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r139 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r140 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r141 storage
(initial 10m next 10m optimal 50m maxextents
```

```
unlimited) tablespace ts_undo;
create public rollback segment r142 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r143 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r144 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r145 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r146 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r147 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r148 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r149 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r150 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r151 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r152 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r153 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r154 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r155 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r156 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r157 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r158 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r159 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
create public rollback segment r160 storage
(initial 10m next 10m optimal 50m maxextents
unlimited) tablespace ts_undo;
}
*wait
# creating extra logfile threads
*sql
{
}
*wait
# building data dictionary
*sql
{
set termout off
set echo off
@E:\oracle814\rdbms\admin\catalog.sql;
@E:\oracle814\rdbms\admin\catparr.sql;
@E:\oracle814\rdbms\admin\catproc.sql;
@E:\oracle814\rdbms\admin\utlxplan.sql;
connect system/manager
@e:\oracle814\dbs\pupbld.sql;
}
*wait
*time=Done database creation
*time=Cycling database before phase 'dbcre'...
*time=Shutting down database...
*sql
{
shutdown normal;
}
*wait
*time=Starting database...
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
}
*wait
*wait
*bgoff
%e-dbcre
```

100g_816_dbgen.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbgen
*bgon=30
################################################
############################
# Data (Flat File) Generation Phase
```

```
*time=Begin creating flat files
*dbgen
{
-v -f -T L -s 100 -C 5 -S 1 -p -i
e:\tpcd814_work\load\item.part -o
i:\flatfiles\lineitem_1
}
*dbgen
{
-v -f -T L -s 100 -C 5 -S 2 -p -i
e:\tpcd814_work\load\item.part -o
i:\flatfiles\lineitem_2
}
*dbgen
{
-v -f -T L -s 100 -C 5 -S 3 -p -i
e:\tpcd814_work\load\item.part -o
i:\flatfiles\lineitem_3
}
*dbgen
{
-v -f -T L -s 100 -C 5 -S 4 -p -i
e:\tpcd814_work\load\item.part -o
i:\flatfiles\lineitem_4
}
*dbgen
{
-v -f -T L -s 100 -C 5 -S 5 -p -i
e:\tpcd814_work\load\item.part -o
i:\flatfiles\lineitem_5
}
*wait
*time=Done creating flat files for table lineitem
*dbgen
{
-v -f -T O -s 100 -C 5 -S 1 -p -i
e:\tpcd814_work\load\ord.part -o
i:\flatfiles\orders_1
}
*dbgen
{
-v -f -T O -s 100 -C 5 -S 2 -p -i
e:\tpcd814_work\load\ord.part -o
i:\flatfiles\orders_2
}
*dbgen
{
-v -f -T O -s 100 -C 5 -S 3 -p -i
e:\tpcd814_work\load\ord.part -o
i:\flatfiles\orders_3
}
*dbgen
{
-v -f -T O -s 100 -C 5 -S 4 -p -i
e:\tpcd814_work\load\ord.part -o
i:\flatfiles\orders_4
}
*dbgen
{
-v -f -T O -s 100 -C 5 -S 5 -p -i
e:\tpcd814_work\load\ord.part -o
i:\flatfiles\orders_5
}
*wait
*time=Done creating flat files for table orders
*dbgen
{
-v -f -T P -s 100 -C 5 -S 1 -p -i
e:\tpcd814_work\load\parts.part -o
i:\flatfiles\parts_1
}
*dbgen
{
-v -f -T P -s 100 -C 5 -S 2 -p -i
e:\tpcd814_work\load\parts.part -o
i:\flatfiles\parts_2
}
*dbgen
{
-v -f -T P -s 100 -C 5 -S 3 -p -i
e:\tpcd814_work\load\parts.part -o
i:\flatfiles\parts_3
}
*dbgen
{
-v -f -T P -s 100 -C 5 -S 4 -p -i
e:\tpcd814_work\load\parts.part -o
i:\flatfiles\parts_4
}
*dbgen
{
-v -f -T P -s 100 -C 5 -S 5 -p -i
e:\tpcd814_work\load\parts.part -o
i:\flatfiles\parts_5
}
*wait
*time=Done creating flat files for table parts
*dbgen
{
-v -f -T c -s 100 -C 5 -S 1 -p -i
e:\tpcd814_work\load\cust.part -o
i:\flatfiles\customer_1
}
*dbgen
{
-v -f -T c -s 100 -C 5 -S 2 -p -i
e:\tpcd814_work\load\cust.part -o
i:\flatfiles\customer_2
}
*dbgen
{
-v -f -T c -s 100 -C 5 -S 3 -p -i
e:\tpcd814_work\load\cust.part -o
i:\flatfiles\customer_3
}
*dbgen
{
-v -f -T c -s 100 -C 5 -S 4 -p -i
e:\tpcd814_work\load\cust.part -o
i:\flatfiles\customer_4
}
*dbgen
{
-v -f -T c -s 100 -C 5 -S 5 -p -i
e:\tpcd814_work\load\cust.part -o
i:\flatfiles\customer_5
}
*wait
*time=Done creating flat files for table customer
*dbgen
{
-v -f -T S -s 100 -C 5 -S 1
}
*dbgen
{
-v -f -T S -s 100 -C 5 -S 2
}
*dbgen
```

```
{
-v -f -T S -s 100 -C 5 -S 3
}
*dbgen
{
-v -f -T S -s 100 -C 5 -S 4
}
*dbgen
{
-v -f -T S -s 100 -C 5 -S 5
}
*wait
*time=Done creating flat files for table partsupp
*dbgen
{
-v -f -T s -s 100 -C 5 -S 1
}
*dbgen
{
-v -f -T s -s 100 -C 5 -S 2
}
*dbgen
{
-v -f -T s -s 100 -C 5 -S 3
}
*dbgen
{
-v -f -T s -s 100 -C 5 -S 4
}
*dbgen
{
-v -f -T s -s 100 -C 5 -S 5
}
*wait
*time=Done creating flat files for table supplier
*dbgen
{
-v -f -T n -s 100 -C 5 -S 1
}
*dbgen
{
-v -f -T n -s 100 -C 5 -S 2
}
*dbgen
{
-v -f -T n -s 100 -C 5 -S 3
}
*dbgen
{
-v -f -T n -s 100 -C 5 -S 4
}
*dbgen
{
-v -f -T n -s 100 -C 5 -S 5
}
*wait
*time=Done creating flat files for table nation
*dbgen
{
-v -f -T r -s 100 -C 5 -S 1
}
*dbgen
{
-v -f -T r -s 100 -C 5 -S 2
}
*dbgen
{
-v -f -T r -s 100 -C 5 -S 3
}
*dbgen
{
-v -f -T r -s 100 -C 5 -S 4
}
*dbgen
{
-v -f -T r -s 100 -C 5 -S 5
}
*wait
*time=Done creating flat files for table region
*wait
*time=Done creating flat files
*wait
*bgoff
%e-dbgen
```

100g_816_ixcre.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-ixcre
*bgon=1
################################################
############################
# Index Creation Phase
*time=Cycling database before phase 'ixcre'...
*time=Shutting down database...
*sql
{
shutdown normal;
}
*wait
*time=Starting database...
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
```

```
}
*wait
*time=Begin dropping constraints
*time=Begin dropping constraint rpk_r_regionkey
*sql
{
connect tpcd/tpcd;
alter table region drop constraint
rpk_r_regionkey;
}
*time=Done dropping constraint rpk_r_regionkey
*time=Begin dropping constraint npk_n_nationkey
*sql
{
connect tpcd/tpcd;
alter table nation drop constraint
npk_n_nationkey;
}
*time=Done dropping constraint npk_n_nationkey
*time=Begin dropping constraint spk_s_suppkey
*sql
{
connect tpcd/tpcd;
alter table supplier drop constraint
spk_s_suppkey;
}
*time=Done dropping constraint spk_s_suppkey
*time=Begin dropping constraint cpk_c_custkey
*sql
{
connect tpcd/tpcd;
alter table customer drop constraint
cpk_c_custkey;
}
*time=Done dropping constraint cpk_c_custkey
*time=Begin dropping constraint ppk_p_partkey
*sql
{
connect tpcd/tpcd;
alter table parts drop constraint ppk_p_partkey;
}
*time=Done dropping constraint ppk_p_partkey
*time=Begin dropping constraint
pspk_ps_partkey_suppkey
*sql
{
connect tpcd/tpcd;
alter table partsupp drop constraint
pspk_ps_partkey_suppkey;
}
*time=Done dropping constraint
pspk_ps_partkey_suppkey
*time=Begin dropping constraint ook_o_orderkey
*sql
{
connect tpcd/tpcd;
alter table orders drop constraint
ook_o_orderkey;
}
*time=Done dropping constraint ook_o_orderkey
*wait
*time=Done dropping constraints
*time=Begin dropping indices
*time=Begin dropping index l_ored
*sql
{
connect tpcd/tpcd;
drop index l_ored;
alter tablespace ts_ind_lored coalesce;
}
*time=Done dropping index l_ored
*time=Begin dropping index o_clokod
*sql
{
connect tpcd/tpcd;
drop index o_clokod;
alter tablespace ts_ind_oclokod coalesce;
}
*time=Done dropping index o_clokod
*time=Begin dropping index o_okey
*sql
{
connect tpcd/tpcd;
drop index o_okey;
alter tablespace ts_ind_ookey coalesce;
}
*time=Done dropping index o_okey
*time=Begin dropping index ij_l_odate
*sql
{
connect tpcd/tpcd;
drop index ij_l_odate;
alter tablespace ts_ij_l_odate coalesce;
}
*time=Done dropping index ij_l_odate
*time=Begin dropping index ij_l_sdate
*sql
{
connect tpcd/tpcd;
drop index ij_l_sdate;
alter tablespace ts_ij_l_sdate coalesce;
}
*time=Done dropping index ij_l_sdate
*time=Begin dropping index ij_l_rdate
*sql
{
connect tpcd/tpcd;
drop index ij_l_rdate;
alter tablespace ts_ij_l_rdate coalesce;
}
*time=Done dropping index ij_l_rdate
*time=Begin dropping index ij_l_pkey
*sql
{
connect tpcd/tpcd;
drop index ij_l_pkey;
alter tablespace ts_ij_l_pkey coalesce;
}
*time=Done dropping index ij_l_pkey
*time=Begin dropping index ij_l_lrid
*sql
{
connect tpcd/tpcd;
drop index ij_l_lrid;
alter tablespace ts_ij_l_lrid coalesce;
}
*time=Done dropping index ij_l_lrid
*time=Begin dropping index ij_l_orid
*sql
{
connect tpcd/tpcd;
drop index ij_l_orid;
alter tablespace ts_ij_l_orid coalesce;
}
*time=Done dropping index ij_l_orid
*time=Begin dropping index ij_l_nullrid
*sql
{
```

```
connect tpcd/tpcd;
drop index ij_l_nullrid;
alter tablespace ts_ij_l_nullrid coalesce;
}
*time=Done dropping index ij_l_nullrid
*time=Begin dropping index ij_p_ppsn
*sql
{
connect tpcd/tpcd;
drop index ij_p_ppsn;
alter tablespace ts_ij_p_ppsn coalesce;
}
*time=Done dropping index ij_p_ppsn
*time=Begin dropping index ij_p_nnpsps
*sql
{
connect tpcd/tpcd;
drop index ij_p_nnpsps;
alter tablespace ts_ij_p_nnpsps coalesce;
}
*time=Done dropping index ij_p_nnpsps
*time=Begin dropping index p_tp
*sql
{
connect tpcd/tpcd;
drop index p_tp;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index p_tp
*time=Begin dropping index p_cbp
*sql
{
connect tpcd/tpcd;
drop index p_cbp;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index p_cbp
*time=Begin dropping index i_n_nationkey
*sql
{
connect tpcd/tpcd;
drop index i_n_nationkey;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index i_n_nationkey
*time=Begin dropping index i_r_regionkey
*sql
{
connect tpcd/tpcd;
drop index i_r_regionkey;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index i_r_regionkey
*time=Begin dropping index i_s_suppkey
*sql
{
connect tpcd/tpcd;
drop index i_s_suppkey;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index i_s_suppkey
*time=Begin dropping index i_c_custkey
*sql
{
connect tpcd/tpcd;
drop index i_c_custkey;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index i_c_custkey
*time=Begin dropping index i_ps_partkey_suppkey
*sql
{
connect tpcd/tpcd;
drop index i_ps_partkey_suppkey;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index i_ps_partkey_suppkey
*time=Begin dropping index i_p_partkey
*sql
{
connect tpcd/tpcd;
drop index i_p_partkey;
alter tablespace ts_mj_ppssrn coalesce;
}
*time=Done dropping index i_p_partkey
*wait
*time=Done dropping indices
*time=Recycling database to ensure freespace
cleanup
*time=Cycling database before phase 'ixcre'...
*time=Shutting down database...
*sql
{
shutdown normal;
}
*wait
*time=Starting database...
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
}
*wait
*time=Begin index creation
*time=Begin creating index l_ored
*sql
{
connect tpcd/tpcd;
create index l_ored
on lineitem
(l_orderkey,l_returnflag,l_extendedprice,l_discou
nt)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ind_lored
storage (initial 8m next 8m maxextents unlimited
pctincrease 0)
parallel (degree 16 )
global partition by range (l_orderkey)
(
partition lored1 values less than (3750001)
,
partition lored2 values less than (7500001)
,
partition lored3 values less than (11250001)
,
partition lored4 values less than (15000001)
,
partition lored5 values less than (18750001)
,
partition lored6 values less than (22500001)
,
partition lored7 values less than (26250001)
,
partition lored8 values less than (30000001)
```

```
,
partition lored9 values less than (33750001)
,
partition lored10 values less than (37500001)
,
partition lored11 values less than (41250001)
,
partition lored12 values less than (45000001)
,
partition lored13 values less than (48750001)
,
partition lored14 values less than (52500001)
,
partition lored15 values less than (56250001)
,
partition lored16 values less than (60000001)
,
partition lored17 values less than (63750001)
,
partition lored18 values less than (67500001)
,
partition lored19 values less than (71250001)
,
partition lored20 values less than (75000001)
,
partition lored21 values less than (78750001)
,
partition lored22 values less than (82500001)
,
partition lored23 values less than (86250001)
,
partition lored24 values less than (90000001)
,
partition lored25 values less than (93750001)
,
partition lored26 values less than (97500001)
,
partition lored27 values less than (101250001)
,
partition lored28 values less than (105000001)
,
partition lored29 values less than (108750001)
,
partition lored30 values less than (112500001)
,
partition lored31 values less than (116250001)
,
partition lored32 values less than (120000001)
,
partition lored33 values less than (123750001)
,
partition lored34 values less than (127500001)
,
partition lored35 values less than (131250001)
,
partition lored36 values less than (135000001)
,
partition lored37 values less than (138750001)
,
partition lored38 values less than (142500001)
,
partition lored39 values less than (146250001)
,
partition lored40 values less than (150000001)
,
partition lored41 values less than (153750001)
,
partition lored42 values less than (157500001)
,
partition lored43 values less than (161250001)
,
partition lored44 values less than (165000001)
,
partition lored45 values less than (168750001)
,
partition lored46 values less than (172500001)
,
partition lored47 values less than (176250001)
,
partition lored48 values less than (180000001)
,
partition lored49 values less than (183750001)
,
partition lored50 values less than (187500001)
,
partition lored51 values less than (191250001)
,
partition lored52 values less than (195000001)
,
partition lored53 values less than (198750001)
,
partition lored54 values less than (202500001)
,
partition lored55 values less than (206250001)
,
partition lored56 values less than (210000001)
,
partition lored57 values less than (213750001)
,
partition lored58 values less than (217500001)
,
partition lored59 values less than (221250001)
,
partition lored60 values less than (225000001)
,
partition lored61 values less than (228750001)
,
partition lored62 values less than (232500001)
,
partition lored63 values less than (236250001)
,
partition lored64 values less than (240000001)
,
partition lored65 values less than (243750001)
,
partition lored66 values less than (247500001)
,
partition lored67 values less than (251250001)
,
partition lored68 values less than (255000001)
,
partition lored69 values less than (258750001)
,
partition lored70 values less than (262500001)
,
partition lored71 values less than (266250001)
,
partition lored72 values less than (270000001)
,
partition lored73 values less than (273750001)
,
partition lored74 values less than (277500001)
,
partition lored75 values less than (281250001)
,
partition lored76 values less than (285000001)
,
partition lored77 values less than (288750001)
```

```
,
partition lored78 values less than (292500001)
,
partition lored79 values less than (296250001)
,
partition lored80 values less than (300000001)
,
partition lored81 values less than (303750001)
,
partition lored82 values less than (307500001)
,
partition lored83 values less than (311250001)
,
partition lored84 values less than (315000001)
,
partition lored85 values less than (318750001)
,
partition lored86 values less than (322500001)
,
partition lored87 values less than (326250001)
,
partition lored88 values less than (330000001)
,
partition lored89 values less than (333750001)
,
partition lored90 values less than (337500001)
,
partition lored91 values less than (341250001)
,
partition lored92 values less than (345000001)
,
partition lored93 values less than (348750001)
,
partition lored94 values less than (352500001)
,
partition lored95 values less than (356250001)
,
partition lored96 values less than (360000001)
,
partition lored97 values less than (363750001)
,
partition lored98 values less than (367500001)
,
partition lored99 values less than (371250001)
,
partition lored100 values less than (375000001)
,
partition lored101 values less than (378750001)
,
partition lored102 values less than (382500001)
,
partition lored103 values less than (386250001)
,
partition lored104 values less than (390000001)
,
partition lored105 values less than (393750001)
,
partition lored106 values less than (397500001)
,
partition lored107 values less than (401250001)
,
partition lored108 values less than (405000001)
,
partition lored109 values less than (408750001)
,
partition lored110 values less than (412500001)
,
partition lored111 values less than (416250001)
,
partition lored112 values less than (420000001)
,
partition lored113 values less than (423750001)
,
partition lored114 values less than (427500001)
,
partition lored115 values less than (431250001)
,
partition lored116 values less than (435000001)
,
partition lored117 values less than (438750001)
,
partition lored118 values less than (442500001)
,
partition lored119 values less than (446250001)
,
partition lored120 values less than (450000001)
,
partition lored121 values less than (453750001)
,
partition lored122 values less than (457500001)
,
partition lored123 values less than (461250001)
,
partition lored124 values less than (465000001)
,
partition lored125 values less than (468750001)
,
partition lored126 values less than (472500001)
,
partition lored127 values less than (476250001)
,
partition lored128 values less than (480000001)
,
partition lored129 values less than (483750001)
,
partition lored130 values less than (487500001)
,
partition lored131 values less than (491250001)
,
partition lored132 values less than (495000001)
,
partition lored133 values less than (498750001)
,
partition lored134 values less than (502500001)
,
partition lored135 values less than (506250001)
,
partition lored136 values less than (510000001)
,
partition lored137 values less than (513750001)
,
partition lored138 values less than (517500001)
,
partition lored139 values less than (521250001)
,
partition lored140 values less than (525000001)
,
partition lored141 values less than (528750001)
,
partition lored142 values less than (532500001)
,
partition lored143 values less than (536250001)
,
partition lored144 values less than (540000001)
,
partition lored145 values less than (543750001)
,
partition lored146 values less than (547500001)
```

```
,
partition lored147 values less than (551250001)
,
partition lored148 values less than (555000001)
,
partition lored149 values less than (558750001)
,
partition lored150 values less than (562500001)
,
partition lored151 values less than (566250001)
,
partition lored152 values less than (570000001)
,
partition lored153 values less than (573750001)
,
partition lored154 values less than (577500001)
,
partition lored155 values less than (581250001)
,
partition lored156 values less than (585000001)
,
partition lored157 values less than (588750001)
,
partition lored158 values less than (592500001)
,
partition lored159 values less than (596250001)
,
partition lored160 values less than (MAXVALUE)
)
;
}
*time=Begin creating index o_clokod
*sql
{
connect tpcd/tpcd;
create index o_clokod
on orders (o_clerk,o_orderkey,o_orderdate)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ind_oclokod
storage (initial 3m next 3m  maxextents unlimited
pctincrease 0)
parallel (degree 16)
local
;
}
*time=Begin creating index o_okey
*sql
{
connect tpcd/tpcd;
create unique index o_okey
on orders (o_orderkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ind_ookey
storage (initial 20m next 20m maxextents
unlimited pctincrease 0)
parallel (degree 16)
;
}
*time=Begin creating index ij_l_odate
*sql
{
connect tpcd/tpcd;
create index ij_l_odate
on mj_locs
(o_orderdate,o_orderkey,l_extendedprice,l_discoun
t,o_ajm,c_ajm,l_commitdate,l_receiptdate,o_orderp
riority,c_custkey,l_returnflag,c_nationkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_odate
storage (initial 14m next 14m maxextents
unlimited pctincrease 0)
parallel (degree 20)
global partition by range (o_orderdate)
(
partition ij_l_odate1 values less than
(to_date('1992-01-16','YYYY-MM-DD'))
,
partition ij_l_odate2 values less than
(to_date('1992-02-01','YYYY-MM-DD'))
,
partition ij_l_odate3 values less than
(to_date('1992-02-16','YYYY-MM-DD'))
,
partition ij_l_odate4 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
,
partition ij_l_odate5 values less than
(to_date('1992-03-16','YYYY-MM-DD'))
,
partition ij_l_odate6 values less than
(to_date('1992-04-01','YYYY-MM-DD'))
,
partition ij_l_odate7 values less than
(to_date('1992-04-16','YYYY-MM-DD'))
,
partition ij_l_odate8 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
,
partition ij_l_odate9 values less than
(to_date('1992-05-16','YYYY-MM-DD'))
,
partition ij_l_odate10 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
,
partition ij_l_odate11 values less than
(to_date('1992-06-16','YYYY-MM-DD'))
,
partition ij_l_odate12 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
,
partition ij_l_odate13 values less than
(to_date('1992-07-16','YYYY-MM-DD'))
,
partition ij_l_odate14 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
,
partition ij_l_odate15 values less than
(to_date('1992-08-16','YYYY-MM-DD'))
,
partition ij_l_odate16 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
,
partition ij_l_odate17 values less than
(to_date('1992-09-16','YYYY-MM-DD'))
,
partition ij_l_odate18 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
,
```

```
partition ij_l_odate19 values less than
(to_date('1992-10-16','YYYY-MM-DD'))
,
partition ij_l_odate20 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
,
partition ij_l_odate21 values less than
(to_date('1992-11-16','YYYY-MM-DD'))
,
partition ij_l_odate22 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
,
partition ij_l_odate23 values less than
(to_date('1992-12-16','YYYY-MM-DD'))
,
partition ij_l_odate24 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
,
partition ij_l_odate25 values less than
(to_date('1993-01-16','YYYY-MM-DD'))
,
partition ij_l_odate26 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
,
partition ij_l_odate27 values less than
(to_date('1993-02-16','YYYY-MM-DD'))
,
partition ij_l_odate28 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
,
partition ij_l_odate29 values less than
(to_date('1993-03-16','YYYY-MM-DD'))
,
partition ij_l_odate30 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
,
partition ij_l_odate31 values less than
(to_date('1993-04-16','YYYY-MM-DD'))
,
partition ij_l_odate32 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
,
partition ij_l_odate33 values less than
(to_date('1993-05-16','YYYY-MM-DD'))
,
partition ij_l_odate34 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
,
partition ij_l_odate35 values less than
(to_date('1993-06-16','YYYY-MM-DD'))
,
partition ij_l_odate36 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
,
partition ij_l_odate37 values less than
(to_date('1993-07-16','YYYY-MM-DD'))
,
partition ij_l_odate38 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
,
partition ij_l_odate39 values less than
(to_date('1993-08-16','YYYY-MM-DD'))
,
partition ij_l_odate40 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
,
partition ij_l_odate41 values less than
(to_date('1993-09-16','YYYY-MM-DD'))
,
partition ij_l_odate42 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
,
partition ij_l_odate43 values less than
(to_date('1993-10-16','YYYY-MM-DD'))
,
partition ij_l_odate44 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
,
partition ij_l_odate45 values less than
(to_date('1993-11-16','YYYY-MM-DD'))
,
partition ij_l_odate46 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
,
partition ij_l_odate47 values less than
(to_date('1993-12-16','YYYY-MM-DD'))
,
partition ij_l_odate48 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
,
partition ij_l_odate49 values less than
(to_date('1994-01-16','YYYY-MM-DD'))
,
partition ij_l_odate50 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
,
partition ij_l_odate51 values less than
(to_date('1994-02-16','YYYY-MM-DD'))
,
partition ij_l_odate52 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
,
partition ij_l_odate53 values less than
(to_date('1994-03-16','YYYY-MM-DD'))
,
partition ij_l_odate54 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
,
partition ij_l_odate55 values less than
(to_date('1994-04-16','YYYY-MM-DD'))
,
partition ij_l_odate56 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
,
partition ij_l_odate57 values less than
(to_date('1994-05-16','YYYY-MM-DD'))
,
partition ij_l_odate58 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
,
partition ij_l_odate59 values less than
(to_date('1994-06-16','YYYY-MM-DD'))
,
partition ij_l_odate60 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
,
partition ij_l_odate61 values less than
(to_date('1994-07-16','YYYY-MM-DD'))
,
partition ij_l_odate62 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
,
partition ij_l_odate63 values less than
(to_date('1994-08-16','YYYY-MM-DD'))
,
partition ij_l_odate64 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
,
```

```
partition ij_l_odate65 values less than
(to_date('1994-09-16','YYYY-MM-DD'))
,
partition ij_l_odate66 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
,
partition ij_l_odate67 values less than
(to_date('1994-10-16','YYYY-MM-DD'))
,
partition ij_l_odate68 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
,
partition ij_l_odate69 values less than
(to_date('1994-11-16','YYYY-MM-DD'))
,
partition ij_l_odate70 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
,
partition ij_l_odate71 values less than
(to_date('1994-12-16','YYYY-MM-DD'))
,
partition ij_l_odate72 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
,
partition ij_l_odate73 values less than
(to_date('1995-01-16','YYYY-MM-DD'))
,
partition ij_l_odate74 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
,
partition ij_l_odate75 values less than
(to_date('1995-02-16','YYYY-MM-DD'))
,
partition ij_l_odate76 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
,
partition ij_l_odate77 values less than
(to_date('1995-03-16','YYYY-MM-DD'))
,
partition ij_l_odate78 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
,
partition ij_l_odate79 values less than
(to_date('1995-04-16','YYYY-MM-DD'))
,
partition ij_l_odate80 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
,
partition ij_l_odate81 values less than
(to_date('1995-05-16','YYYY-MM-DD'))
,
partition ij_l_odate82 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
,
partition ij_l_odate83 values less than
(to_date('1995-06-16','YYYY-MM-DD'))
,
partition ij_l_odate84 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
,
partition ij_l_odate85 values less than
(to_date('1995-07-16','YYYY-MM-DD'))
,
partition ij_l_odate86 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
,
partition ij_l_odate87 values less than
(to_date('1995-08-16','YYYY-MM-DD'))
,
partition ij_l_odate88 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
,
partition ij_l_odate89 values less than
(to_date('1995-09-16','YYYY-MM-DD'))
,
partition ij_l_odate90 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
,
partition ij_l_odate91 values less than
(to_date('1995-10-16','YYYY-MM-DD'))
,
partition ij_l_odate92 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
,
partition ij_l_odate93 values less than
(to_date('1995-11-16','YYYY-MM-DD'))
,
partition ij_l_odate94 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
,
partition ij_l_odate95 values less than
(to_date('1995-12-16','YYYY-MM-DD'))
,
partition ij_l_odate96 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
,
partition ij_l_odate97 values less than
(to_date('1996-01-16','YYYY-MM-DD'))
,
partition ij_l_odate98 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
,
partition ij_l_odate99 values less than
(to_date('1996-02-16','YYYY-MM-DD'))
,
partition ij_l_odate100 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
,
partition ij_l_odate101 values less than
(to_date('1996-03-16','YYYY-MM-DD'))
,
partition ij_l_odate102 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
,
partition ij_l_odate103 values less than
(to_date('1996-04-16','YYYY-MM-DD'))
,
partition ij_l_odate104 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
,
partition ij_l_odate105 values less than
(to_date('1996-05-16','YYYY-MM-DD'))
,
partition ij_l_odate106 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
,
partition ij_l_odate107 values less than
(to_date('1996-06-16','YYYY-MM-DD'))
,
partition ij_l_odate108 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
,
partition ij_l_odate109 values less than
(to_date('1996-07-16','YYYY-MM-DD'))
,
partition ij_l_odate110 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
,
```

```
partition ij_l_odate111 values less than
(to_date('1996-08-16','YYYY-MM-DD'))
,
partition ij_l_odate112 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
,
partition ij_l_odate113 values less than
(to_date('1996-09-16','YYYY-MM-DD'))
,
partition ij_l_odate114 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
,
partition ij_l_odate115 values less than
(to_date('1996-10-16','YYYY-MM-DD'))
,
partition ij_l_odate116 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
,
partition ij_l_odate117 values less than
(to_date('1996-11-16','YYYY-MM-DD'))
,
partition ij_l_odate118 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
,
partition ij_l_odate119 values less than
(to_date('1996-12-16','YYYY-MM-DD'))
,
partition ij_l_odate120 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
,
partition ij_l_odate121 values less than
(to_date('1997-01-16','YYYY-MM-DD'))
,
partition ij_l_odate122 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
,
partition ij_l_odate123 values less than
(to_date('1997-02-16','YYYY-MM-DD'))
,
partition ij_l_odate124 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
,
partition ij_l_odate125 values less than
(to_date('1997-03-16','YYYY-MM-DD'))
,
partition ij_l_odate126 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
,
partition ij_l_odate127 values less than
(to_date('1997-04-16','YYYY-MM-DD'))
,
partition ij_l_odate128 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
,
partition ij_l_odate129 values less than
(to_date('1997-05-16','YYYY-MM-DD'))
,
partition ij_l_odate130 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
,
partition ij_l_odate131 values less than
(to_date('1997-06-16','YYYY-MM-DD'))
,
partition ij_l_odate132 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
,
partition ij_l_odate133 values less than
(to_date('1997-07-16','YYYY-MM-DD'))
,
partition ij_l_odate134 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
,
partition ij_l_odate135 values less than
(to_date('1997-08-16','YYYY-MM-DD'))
,
partition ij_l_odate136 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
,
partition ij_l_odate137 values less than
(to_date('1997-09-16','YYYY-MM-DD'))
,
partition ij_l_odate138 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
,
partition ij_l_odate139 values less than
(to_date('1997-10-16','YYYY-MM-DD'))
,
partition ij_l_odate140 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
,
partition ij_l_odate141 values less than
(to_date('1997-11-16','YYYY-MM-DD'))
,
partition ij_l_odate142 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
,
partition ij_l_odate143 values less than
(to_date('1997-12-16','YYYY-MM-DD'))
,
partition ij_l_odate144 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
,
partition ij_l_odate145 values less than
(to_date('1998-01-16','YYYY-MM-DD'))
,
partition ij_l_odate146 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
,
partition ij_l_odate147 values less than
(to_date('1998-02-16','YYYY-MM-DD'))
,
partition ij_l_odate148 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
,
partition ij_l_odate149 values less than
(to_date('1998-03-16','YYYY-MM-DD'))
,
partition ij_l_odate150 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
,
partition ij_l_odate151 values less than
(to_date('1998-04-16','YYYY-MM-DD'))
,
partition ij_l_odate152 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
,
partition ij_l_odate153 values less than
(to_date('1998-05-16','YYYY-MM-DD'))
,
partition ij_l_odate154 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
,
partition ij_l_odate155 values less than
(to_date('1998-06-16','YYYY-MM-DD'))
,
partition ij_l_odate156 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
,
```

```
partition ij_l_odate157 values less than
(to_date('1998-07-16','YYYY-MM-DD'))
,
partition ij_l_odate158 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
,
partition ij_l_odate159 values less than
(to_date('1998-08-16','YYYY-MM-DD'))
,
partition ij_l_odate160 values less than
(MAXVALUE)
)
;
}
*time=Begin creating index ij_l_sdate
*sql
{
connect tpcd/tpcd;
create index ij_l_sdate
on mj_locs
(o_orderkey,o_orderdate,o_ajm,c_ajm,l_extendedpri
ce,l_discount,c_mktsegment,l_shipdate,l_orderkey,
l_partkey,o_shippriority,l_suppkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_sdate
storage (initial 20m next 20m maxextents
unlimited pctincrease 0)
parallel (degree 16)
local
;
}
*time=Begin creating index ij_l_rdate
*sql
{
connect tpcd/tpcd;
create index ij_l_rdate
on mj_locs
(l_receiptdate,o_orderkey,o_ajm,l_shipdate,l_ship
mode,l_commitdate,o_orderpriority)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_rdate
storage (initial 10m next 10m maxextents
unlimited pctincrease 0)
parallel (degree 20)
global partition by range (l_receiptdate)
(
partition ij_l_rdate1 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate2 values less than
(to_date('1992-03-24','YYYY-MM-DD'))
,
partition ij_l_rdate3 values less than
(to_date('1992-04-14','YYYY-MM-DD'))
,
partition ij_l_rdate4 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate5 values less than
(to_date('1992-05-16','YYYY-MM-DD'))
,
partition ij_l_rdate6 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate7 values less than
(to_date('1992-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate8 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate9 values less than
(to_date('1992-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate10 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate11 values less than
(to_date('1992-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate12 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate13 values less than
(to_date('1992-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate14 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate15 values less than
(to_date('1992-10-16','YYYY-MM-DD'))
,
partition ij_l_rdate16 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate17 values less than
(to_date('1992-11-16','YYYY-MM-DD'))
,
partition ij_l_rdate18 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
,
partition ij_l_rdate19 values less than
(to_date('1992-12-16','YYYY-MM-DD'))
,
partition ij_l_rdate20 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
,
partition ij_l_rdate21 values less than
(to_date('1993-01-16','YYYY-MM-DD'))
,
partition ij_l_rdate22 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
,
partition ij_l_rdate23 values less than
(to_date('1993-02-16','YYYY-MM-DD'))
,
partition ij_l_rdate24 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate25 values less than
(to_date('1993-03-16','YYYY-MM-DD'))
,
partition ij_l_rdate26 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
,
partition ij_l_rdate27 values less than
(to_date('1993-04-16','YYYY-MM-DD'))
,
partition ij_l_rdate28 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate29 values less than
(to_date('1993-05-16','YYYY-MM-DD'))
```

```
,
partition ij_l_rdate30 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate31 values less than
(to_date('1993-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate32 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate33 values less than
(to_date('1993-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate34 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate35 values less than
(to_date('1993-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate36 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate37 values less than
(to_date('1993-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate38 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate39 values less than
(to_date('1993-10-16','YYYY-MM-DD'))
,
partition ij_l_rdate40 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate41 values less than
(to_date('1993-11-16','YYYY-MM-DD'))
,
partition ij_l_rdate42 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
,
partition ij_l_rdate43 values less than
(to_date('1993-12-16','YYYY-MM-DD'))
,
partition ij_l_rdate44 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
,
partition ij_l_rdate45 values less than
(to_date('1994-01-16','YYYY-MM-DD'))
,
partition ij_l_rdate46 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
,
partition ij_l_rdate47 values less than
(to_date('1994-02-16','YYYY-MM-DD'))
,
partition ij_l_rdate48 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate49 values less than
(to_date('1994-03-16','YYYY-MM-DD'))
,
partition ij_l_rdate50 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
,
partition ij_l_rdate51 values less than
(to_date('1994-04-16','YYYY-MM-DD'))
,
partition ij_l_rdate52 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate53 values less than
(to_date('1994-05-16','YYYY-MM-DD'))
,
partition ij_l_rdate54 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate55 values less than
(to_date('1994-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate56 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate57 values less than
(to_date('1994-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate58 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate59 values less than
(to_date('1994-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate60 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate61 values less than
(to_date('1994-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate62 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate63 values less than
(to_date('1994-10-16','YYYY-MM-DD'))
,
partition ij_l_rdate64 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate65 values less than
(to_date('1994-11-16','YYYY-MM-DD'))
,
partition ij_l_rdate66 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
,
partition ij_l_rdate67 values less than
(to_date('1994-12-16','YYYY-MM-DD'))
,
partition ij_l_rdate68 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
,
partition ij_l_rdate69 values less than
(to_date('1995-01-16','YYYY-MM-DD'))
,
partition ij_l_rdate70 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
,
partition ij_l_rdate71 values less than
(to_date('1995-02-16','YYYY-MM-DD'))
,
partition ij_l_rdate72 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate73 values less than
(to_date('1995-03-16','YYYY-MM-DD'))
,
partition ij_l_rdate74 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
,
partition ij_l_rdate75 values less than
(to_date('1995-04-16','YYYY-MM-DD'))
```

```
,
partition ij_l_rdate76 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate77 values less than
(to_date('1995-05-16','YYYY-MM-DD'))
,
partition ij_l_rdate78 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate79 values less than
(to_date('1995-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate80 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate81 values less than
(to_date('1995-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate82 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate83 values less than
(to_date('1995-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate84 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate85 values less than
(to_date('1995-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate86 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate87 values less than
(to_date('1995-10-16','YYYY-MM-DD'))
,
partition ij_l_rdate88 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate89 values less than
(to_date('1995-11-16','YYYY-MM-DD'))
,
partition ij_l_rdate90 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
,
partition ij_l_rdate91 values less than
(to_date('1995-12-16','YYYY-MM-DD'))
,
partition ij_l_rdate92 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
,
partition ij_l_rdate93 values less than
(to_date('1996-01-16','YYYY-MM-DD'))
,
partition ij_l_rdate94 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
,
partition ij_l_rdate95 values less than
(to_date('1996-02-16','YYYY-MM-DD'))
,
partition ij_l_rdate96 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate97 values less than
(to_date('1996-03-16','YYYY-MM-DD'))
,
partition ij_l_rdate98 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
,
partition ij_l_rdate99 values less than
(to_date('1996-04-16','YYYY-MM-DD'))
,
partition ij_l_rdate100 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate101 values less than
(to_date('1996-05-16','YYYY-MM-DD'))
,
partition ij_l_rdate102 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate103 values less than
(to_date('1996-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate104 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate105 values less than
(to_date('1996-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate106 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate107 values less than
(to_date('1996-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate108 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate109 values less than
(to_date('1996-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate110 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate111 values less than
(to_date('1996-10-16','YYYY-MM-DD'))
,
partition ij_l_rdate112 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate113 values less than
(to_date('1996-11-16','YYYY-MM-DD'))
,
partition ij_l_rdate114 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
,
partition ij_l_rdate115 values less than
(to_date('1996-12-16','YYYY-MM-DD'))
,
partition ij_l_rdate116 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
,
partition ij_l_rdate117 values less than
(to_date('1997-01-16','YYYY-MM-DD'))
,
partition ij_l_rdate118 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
,
partition ij_l_rdate119 values less than
(to_date('1997-02-16','YYYY-MM-DD'))
,
partition ij_l_rdate120 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate121 values less than
(to_date('1997-03-16','YYYY-MM-DD'))
```

```
,
partition ij_l_rdate122 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
,
partition ij_l_rdate123 values less than
(to_date('1997-04-16','YYYY-MM-DD'))
,
partition ij_l_rdate124 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate125 values less than
(to_date('1997-05-16','YYYY-MM-DD'))
,
partition ij_l_rdate126 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate127 values less than
(to_date('1997-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate128 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate129 values less than
(to_date('1997-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate130 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate131 values less than
(to_date('1997-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate132 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate133 values less than
(to_date('1997-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate134 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate135 values less than
(to_date('1997-10-16','YYYY-MM-DD'))
,
partition ij_l_rdate136 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate137 values less than
(to_date('1997-11-16','YYYY-MM-DD'))
,
partition ij_l_rdate138 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
,
partition ij_l_rdate139 values less than
(to_date('1997-12-16','YYYY-MM-DD'))
,
partition ij_l_rdate140 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
,
partition ij_l_rdate141 values less than
(to_date('1998-01-16','YYYY-MM-DD'))
,
partition ij_l_rdate142 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
,
partition ij_l_rdate143 values less than
(to_date('1998-02-16','YYYY-MM-DD'))
,
partition ij_l_rdate144 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
,
partition ij_l_rdate145 values less than
(to_date('1998-03-16','YYYY-MM-DD'))
,
partition ij_l_rdate146 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
,
partition ij_l_rdate147 values less than
(to_date('1998-04-16','YYYY-MM-DD'))
,
partition ij_l_rdate148 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
,
partition ij_l_rdate149 values less than
(to_date('1998-05-16','YYYY-MM-DD'))
,
partition ij_l_rdate150 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
,
partition ij_l_rdate151 values less than
(to_date('1998-06-16','YYYY-MM-DD'))
,
partition ij_l_rdate152 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
,
partition ij_l_rdate153 values less than
(to_date('1998-07-16','YYYY-MM-DD'))
,
partition ij_l_rdate154 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
,
partition ij_l_rdate155 values less than
(to_date('1998-08-16','YYYY-MM-DD'))
,
partition ij_l_rdate156 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
,
partition ij_l_rdate157 values less than
(to_date('1998-09-16','YYYY-MM-DD'))
,
partition ij_l_rdate158 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
,
partition ij_l_rdate159 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
,
partition ij_l_rdate160 values less than
(MAXVALUE)
)
;
}
*time=Begin creating index ij_l_pkey
*sql
{
connect tpcd/tpcd;
create index ij_l_pkey
on mj_locs
(l_partkey,o_orderkey,l_extendedprice,l_discount,
l_quantity,o_orderdate,o_ajm,c_ajm,s_ajm,l_suppke
y,s_suppkey,c_nationkey,s_nationkey)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_pkey
storage (initial 8m next 8m freelists 32
maxextents unlimited pctincrease 0)
parallel (degree 20)
global partition by range (l_partkey)
```

```
(
partition ij_l_pkey1 values less than (125001)
,
partition ij_l_pkey2 values less than (250001)
,
partition ij_l_pkey3 values less than (375001)
,
partition ij_l_pkey4 values less than (500001)
,
partition ij_l_pkey5 values less than (625001)
,
partition ij_l_pkey6 values less than (750001)
,
partition ij_l_pkey7 values less than (875001)
,
partition ij_l_pkey8 values less than (1000001)
,
partition ij_l_pkey9 values less than (1125001)
,
partition ij_l_pkey10 values less than (1250001)
,
partition ij_l_pkey11 values less than (1375001)
,
partition ij_l_pkey12 values less than (1500001)
,
partition ij_l_pkey13 values less than (1625001)
,
partition ij_l_pkey14 values less than (1750001)
,
partition ij_l_pkey15 values less than (1875001)
,
partition ij_l_pkey16 values less than (2000001)
,
partition ij_l_pkey17 values less than (2125001)
,
partition ij_l_pkey18 values less than (2250001)
,
partition ij_l_pkey19 values less than (2375001)
,
partition ij_l_pkey20 values less than (2500001)
,
partition ij_l_pkey21 values less than (2625001)
,
partition ij_l_pkey22 values less than (2750001)
,
partition ij_l_pkey23 values less than (2875001)
,
partition ij_l_pkey24 values less than (3000001)
,
partition ij_l_pkey25 values less than (3125001)
,
partition ij_l_pkey26 values less than (3250001)
,
partition ij_l_pkey27 values less than (3375001)
,
partition ij_l_pkey28 values less than (3500001)
,
partition ij_l_pkey29 values less than (3625001)
,
partition ij_l_pkey30 values less than (3750001)
,
partition ij_l_pkey31 values less than (3875001)
,
partition ij_l_pkey32 values less than (4000001)
,
partition ij_l_pkey33 values less than (4125001)
,
partition ij_l_pkey34 values less than (4250001)
,
partition ij_l_pkey35 values less than (4375001)
,
partition ij_l_pkey36 values less than (4500001)
,
partition ij_l_pkey37 values less than (4625001)
,
partition ij_l_pkey38 values less than (4750001)
,
partition ij_l_pkey39 values less than (4875001)
,
partition ij_l_pkey40 values less than (5000001)
,
partition ij_l_pkey41 values less than (5125001)
,
partition ij_l_pkey42 values less than (5250001)
,
partition ij_l_pkey43 values less than (5375001)
,
partition ij_l_pkey44 values less than (5500001)
,
partition ij_l_pkey45 values less than (5625001)
,
partition ij_l_pkey46 values less than (5750001)
,
partition ij_l_pkey47 values less than (5875001)
,
partition ij_l_pkey48 values less than (6000001)
,
partition ij_l_pkey49 values less than (6125001)
,
partition ij_l_pkey50 values less than (6250001)
,
partition ij_l_pkey51 values less than (6375001)
,
partition ij_l_pkey52 values less than (6500001)
,
partition ij_l_pkey53 values less than (6625001)
,
partition ij_l_pkey54 values less than (6750001)
,
partition ij_l_pkey55 values less than (6875001)
,
partition ij_l_pkey56 values less than (7000001)
,
partition ij_l_pkey57 values less than (7125001)
,
partition ij_l_pkey58 values less than (7250001)
,
partition ij_l_pkey59 values less than (7375001)
,
partition ij_l_pkey60 values less than (7500001)
,
partition ij_l_pkey61 values less than (7625001)
,
partition ij_l_pkey62 values less than (7750001)
,
partition ij_l_pkey63 values less than (7875001)
,
partition ij_l_pkey64 values less than (8000001)
,
partition ij_l_pkey65 values less than (8125001)
,
partition ij_l_pkey66 values less than (8250001)
,
partition ij_l_pkey67 values less than (8375001)
,
partition ij_l_pkey68 values less than (8500001)
,
partition ij_l_pkey69 values less than (8625001)
```

```
,
partition ij_l_pkey70 values less than (8750001)
,
partition ij_l_pkey71 values less than (8875001)
,
partition ij_l_pkey72 values less than (9000001)
,
partition ij_l_pkey73 values less than (9125001)
,
partition ij_l_pkey74 values less than (9250001)
,
partition ij_l_pkey75 values less than (9375001)
,
partition ij_l_pkey76 values less than (9500001)
,
partition ij_l_pkey77 values less than (9625001)
,
partition ij_l_pkey78 values less than (9750001)
,
partition ij_l_pkey79 values less than (9875001)
,
partition ij_l_pkey80 values less than (10000001)
,
partition ij_l_pkey81 values less than (10125001)
,
partition ij_l_pkey82 values less than (10250001)
,
partition ij_l_pkey83 values less than (10375001)
,
partition ij_l_pkey84 values less than (10500001)
,
partition ij_l_pkey85 values less than (10625001)
,
partition ij_l_pkey86 values less than (10750001)
,
partition ij_l_pkey87 values less than (10875001)
,
partition ij_l_pkey88 values less than (11000001)
,
partition ij_l_pkey89 values less than (11125001)
,
partition ij_l_pkey90 values less than (11250001)
,
partition ij_l_pkey91 values less than (11375001)
,
partition ij_l_pkey92 values less than (11500001)
,
partition ij_l_pkey93 values less than (11625001)
,
partition ij_l_pkey94 values less than (11750001)
,
partition ij_l_pkey95 values less than (11875001)
,
partition ij_l_pkey96 values less than (12000001)
,
partition ij_l_pkey97 values less than (12125001)
,
partition ij_l_pkey98 values less than (12250001)
,
partition ij_l_pkey99 values less than (12375001)
,
partition ij_l_pkey100 values less than
(12500001)
,
partition ij_l_pkey101 values less than
(12625001)
,
partition ij_l_pkey102 values less than
(12750001)
,
partition ij_l_pkey103 values less than
(12875001)
,
partition ij_l_pkey104 values less than
(13000001)
,
partition ij_l_pkey105 values less than
(13125001)
,
partition ij_l_pkey106 values less than
(13250001)
,
partition ij_l_pkey107 values less than
(13375001)
,
partition ij_l_pkey108 values less than
(13500001)
,
partition ij_l_pkey109 values less than
(13625001)
,
partition ij_l_pkey110 values less than
(13750001)
,
partition ij_l_pkey111 values less than
(13875001)
,
partition ij_l_pkey112 values less than
(14000001)
,
partition ij_l_pkey113 values less than
(14125001)
,
partition ij_l_pkey114 values less than
(14250001)
,
partition ij_l_pkey115 values less than
(14375001)
,
partition ij_l_pkey116 values less than
(14500001)
,
partition ij_l_pkey117 values less than
(14625001)
,
partition ij_l_pkey118 values less than
(14750001)
,
partition ij_l_pkey119 values less than
(14875001)
,
partition ij_l_pkey120 values less than
(15000001)
,
partition ij_l_pkey121 values less than
(15125001)
,
partition ij_l_pkey122 values less than
(15250001)
,
partition ij_l_pkey123 values less than
(15375001)
,
partition ij_l_pkey124 values less than
(15500001)
,
partition ij_l_pkey125 values less than
(15625001)
```

```
,
partition ij_l_pkey126 values less than
(15750001)
,
partition ij_l_pkey127 values less than
(15875001)
,
partition ij_l_pkey128 values less than
(16000001)
,
partition ij_l_pkey129 values less than
(16125001)
,
partition ij_l_pkey130 values less than
(16250001)
,
partition ij_l_pkey131 values less than
(16375001)
,
partition ij_l_pkey132 values less than
(16500001)
,
partition ij_l_pkey133 values less than
(16625001)
,
partition ij_l_pkey134 values less than
(16750001)
,
partition ij_l_pkey135 values less than
(16875001)
,
partition ij_l_pkey136 values less than
(17000001)
,
partition ij_l_pkey137 values less than
(17125001)
,
partition ij_l_pkey138 values less than
(17250001)
,
partition ij_l_pkey139 values less than
(17375001)
,
partition ij_l_pkey140 values less than
(17500001)
,
partition ij_l_pkey141 values less than
(17625001)
,
partition ij_l_pkey142 values less than
(17750001)
,
partition ij_l_pkey143 values less than
(17875001)
,
partition ij_l_pkey144 values less than
(18000001)
,
partition ij_l_pkey145 values less than
(18125001)
,
partition ij_l_pkey146 values less than
(18250001)
,
partition ij_l_pkey147 values less than
(18375001)
,
partition ij_l_pkey148 values less than
(18500001)
,
partition ij_l_pkey149 values less than
(18625001)
,
partition ij_l_pkey150 values less than
(18750001)
,
partition ij_l_pkey151 values less than
(18875001)
,
partition ij_l_pkey152 values less than
(19000001)
,
partition ij_l_pkey153 values less than
(19125001)
,
partition ij_l_pkey154 values less than
(19250001)
,
partition ij_l_pkey155 values less than
(19375001)
,
partition ij_l_pkey156 values less than
(19500001)
,
partition ij_l_pkey157 values less than
(19625001)
,
partition ij_l_pkey158 values less than
(19750001)
,
partition ij_l_pkey159 values less than
(19875001)
,
partition ij_l_pkey160 values less than
(MAXVALUE)
)
;
}
*time=Begin creating index ij_l_lrid
*sql
{
connect tpcd/tpcd;
create index ij_l_lrid
on mj_locs (l_rowid)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_lrid
storage (initial 100m next 100m maxextents
unlimited pctincrease 0)
parallel (degree 16)
;
}
*time=Begin creating index ij_l_orid
*sql
{
connect tpcd/tpcd;
create index ij_l_orid
on mj_locs (o_rowid)
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_orid
storage (initial 100m next 100m maxextents
unlimited pctincrease 0)
parallel (degree 16)
```

```
;
}
*time=Begin creating index ij_l_nullrid
*sql
{
connect tpcd/tpcd;
create index ij_l_nullrid
on mj_locs (decode(o_rowid, NULL, 1, NULL))
pctfree 2
initrans 10
nologging
compute statistics
tablespace ts_ij_l_nullrid
storage (initial 8m next 8m maxextents unlimited
pctincrease 0)
parallel (degree 20)
;
}
*time=Begin creating index ij_p_ppsn
*sql
{
connect tpcd/tpcd;
create index ij_p_ppsn
on mj_ppssrn
(p_partkey,ps_partkey,ps_supplycost,n_ajm,r_ajm,r
_name,s_ajm)
pctfree 0
nologging
compute statistics
tablespace ts_ij_p_ppsn
storage (initial 2m next 2m maxextents unlimited
pctincrease 0)
parallel (degree 20)
local
;
}
*time=Begin creating index ij_p_nnpsps
*sql
{
connect tpcd/tpcd;
create index ij_p_nnpsps
on mj_ppssrn
(n_name,n_ajm,s_ajm,ps_partkey,ps_supplycost,ps_a
vailqty)
pctfree 0
nologging
compute statistics
tablespace ts_ij_p_nnpsps
storage (initial 7m next 7m maxextents unlimited
pctincrease 0)
parallel (degree 16)
global partition by range (n_name)
(
partition ij_p_nnpsps1 values less than
('ALGERIB')
,
partition ij_p_nnpsps2 values less than
('ARGENTINB')
,
partition ij_p_nnpsps3 values less than
('BRAZIM')
,
partition ij_p_nnpsps4 values less than
('CANADB')
,
partition ij_p_nnpsps5 values less than ('CHINB')
,
partition ij_p_nnpsps6 values less than ('EGYPY')
,
partition ij_p_nnpsps7 values less than
('ETHIOPIB')
,
partition ij_p_nnpsps8 values less than
('FRANCF')
,
partition ij_p_nnpsps9 values less than
('GERMANZ')
,
partition ij_p_nnpsps10 values less than
('INDIB')
,
partition ij_p_nnpsps11 values less than
('INDONESIB')
,
partition ij_p_nnpsps12 values less than ('IRAO')
,
partition ij_p_nnpsps13 values less than ('IRAR')
,
partition ij_p_nnpsps14 values less than
('JAPAO')
,
partition ij_p_nnpsps15 values less than
('JORDAO')
,
partition ij_p_nnpsps16 values less than
('KENYB')
,
partition ij_p_nnpsps17 values less than
('MOROCCP')
,
partition ij_p_nnpsps18 values less than
('MOZAMBIQUF')
,
partition ij_p_nnpsps19 values less than ('PERV')
,
partition ij_p_nnpsps20 values less than
('ROMANIB')
,
partition ij_p_nnpsps21 values less than
('RUSSIB')
,
partition ij_p_nnpsps22 values less than ('SAUDI
ARABIB')
,
partition ij_p_nnpsps23 values less than ('UNITED
KINGDON')
,
partition ij_p_nnpsps24 values less than
('VIETNAN')
,
partition ij_p_nnpsps25 values less than
(MAXVALUE)
)
;
}
*time=Begin creating index p_tp
*sql
{
connect tpcd/tpcd;
create index p_tp
on parts (p_type,p_partkey)
pctfree 1
nologging
compute statistics
tablespace ts_mj_ppssrn
storage (initial 4m next 4m maxextents unlimited
pctincrease 0)
parallel (degree 16)
```

```
global partition by range (p_type)
(
partition ptp1 values less than ('ECONOMY
ANODIZED COPPER')
,
partition ptp2 values less than ('ECONOMY
ANODIZED NICKEL')
,
partition ptp3 values less than ('ECONOMY
ANODIZED STEEL')
,
partition ptp4 values less than ('ECONOMY
ANODIZED TIN')
,
partition ptp5 values less than ('ECONOMY BRUSHED
BRASS')
,
partition ptp6 values less than ('ECONOMY BRUSHED
COPPER')
,
partition ptp7 values less than ('ECONOMY BRUSHED
NICKEL')
,
partition ptp8 values less than ('ECONOMY BRUSHED
STEEL')
,
partition ptp9 values less than ('ECONOMY BRUSHED
TIN')
,
partition ptp10 values less than ('ECONOMY
BURNISHED BRASS')
,
partition ptp11 values less than ('ECONOMY
BURNISHED COPPER')
,
partition ptp12 values less than ('ECONOMY
BURNISHED NICKEL')
,
partition ptp13 values less than ('ECONOMY
BURNISHED STEEL')
,
partition ptp14 values less than ('ECONOMY
BURNISHED TIN')
,
partition ptp15 values less than ('ECONOMY PLATED
BRASS')
,
partition ptp16 values less than ('ECONOMY PLATED
COPPER')
,
partition ptp17 values less than ('ECONOMY PLATED
NICKEL')
,
partition ptp18 values less than ('ECONOMY PLATED
STEEL')
,
partition ptp19 values less than ('ECONOMY PLATED
TIN')
,
partition ptp20 values less than ('ECONOMY
POLISHED BRASS')
,
partition ptp21 values less than ('ECONOMY
POLISHED COPPER')
,
partition ptp22 values less than ('ECONOMY
POLISHED NICKEL')
,
partition ptp23 values less than ('ECONOMY
POLISHED STEEL')
,
partition ptp24 values less than ('ECONOMY
POLISHED TIN')
,
partition ptp25 values less than ('LARGE ANODIZED
BRASS')
,
partition ptp26 values less than ('LARGE ANODIZED
COPPER')
,
partition ptp27 values less than ('LARGE ANODIZED
NICKEL')
,
partition ptp28 values less than ('LARGE ANODIZED
STEEL')
,
partition ptp29 values less than ('LARGE ANODIZED
TIN')
,
partition ptp30 values less than ('LARGE BRUSHED
BRASS')
,
partition ptp31 values less than ('LARGE BRUSHED
COPPER')
,
partition ptp32 values less than ('LARGE BRUSHED
NICKEL')
,
partition ptp33 values less than ('LARGE BRUSHED
STEEL')
,
partition ptp34 values less than ('LARGE BRUSHED
TIN')
,
partition ptp35 values less than ('LARGE
BURNISHED BRASS')
,
partition ptp36 values less than ('LARGE
BURNISHED COPPER')
,
partition ptp37 values less than ('LARGE
BURNISHED NICKEL')
,
partition ptp38 values less than ('LARGE
BURNISHED STEEL')
,
partition ptp39 values less than ('LARGE
BURNISHED TIN')
,
partition ptp40 values less than ('LARGE PLATED
BRASS')
,
partition ptp41 values less than ('LARGE PLATED
COPPER')
,
partition ptp42 values less than ('LARGE PLATED
NICKEL')
,
partition ptp43 values less than ('LARGE PLATED
STEEL')
,
partition ptp44 values less than ('LARGE PLATED
TIN')
,
partition ptp45 values less than ('LARGE POLISHED
BRASS')
,
```

```
partition ptp46 values less than ('LARGE POLISHED
COPPER')
,
partition ptp47 values less than ('LARGE POLISHED
NICKEL')
,
partition ptp48 values less than ('LARGE POLISHED
STEEL')
,
partition ptp49 values less than ('LARGE POLISHED
TIN')
,
partition ptp50 values less than ('MEDIUM
ANODIZED BRASS')
,
partition ptp51 values less than ('MEDIUM
ANODIZED COPPER')
,
partition ptp52 values less than ('MEDIUM
ANODIZED NICKEL')
,
partition ptp53 values less than ('MEDIUM
ANODIZED STEEL')
,
partition ptp54 values less than ('MEDIUM
ANODIZED TIN')
,
partition ptp55 values less than ('MEDIUM BRUSHED
BRASS')
,
partition ptp56 values less than ('MEDIUM BRUSHED
COPPER')
,
partition ptp57 values less than ('MEDIUM BRUSHED
NICKEL')
,
partition ptp58 values less than ('MEDIUM BRUSHED
STEEL')
,
partition ptp59 values less than ('MEDIUM BRUSHED
TIN')
,
partition ptp60 values less than ('MEDIUM
BURNISHED BRASS')
,
partition ptp61 values less than ('MEDIUM
BURNISHED COPPER')
,
partition ptp62 values less than ('MEDIUM
BURNISHED NICKEL')
,
partition ptp63 values less than ('MEDIUM
BURNISHED STEEL')
,
partition ptp64 values less than ('MEDIUM
BURNISHED TIN')
,
partition ptp65 values less than ('MEDIUM PLATED
BRASS')
,
partition ptp66 values less than ('MEDIUM PLATED
COPPER')
,
partition ptp67 values less than ('MEDIUM PLATED
NICKEL')
,
partition ptp68 values less than ('MEDIUM PLATED
STEEL')
,
partition ptp69 values less than ('MEDIUM PLATED
TIN')
,
partition ptp70 values less than ('MEDIUM
POLISHED BRASS')
,
partition ptp71 values less than ('MEDIUM
POLISHED COPPER')
,
partition ptp72 values less than ('MEDIUM
POLISHED NICKEL')
,
partition ptp73 values less than ('MEDIUM
POLISHED STEEL')
,
partition ptp74 values less than ('MEDIUM
POLISHED TIN')
,
partition ptp75 values less than ('PROMO ANODIZED
BRASS')
,
partition ptp76 values less than ('PROMO ANODIZED
COPPER')
,
partition ptp77 values less than ('PROMO ANODIZED
NICKEL')
,
partition ptp78 values less than ('PROMO ANODIZED
STEEL')
,
partition ptp79 values less than ('PROMO ANODIZED
TIN')
,
partition ptp80 values less than ('PROMO BRUSHED
BRASS')
,
partition ptp81 values less than ('PROMO BRUSHED
COPPER')
,
partition ptp82 values less than ('PROMO BRUSHED
NICKEL')
,
partition ptp83 values less than ('PROMO BRUSHED
STEEL')
,
partition ptp84 values less than ('PROMO BRUSHED
TIN')
,
partition ptp85 values less than ('PROMO
BURNISHED BRASS')
,
partition ptp86 values less than ('PROMO
BURNISHED COPPER')
,
partition ptp87 values less than ('PROMO
BURNISHED NICKEL')
,
partition ptp88 values less than ('PROMO
BURNISHED STEEL')
,
partition ptp89 values less than ('PROMO
BURNISHED TIN')
,
partition ptp90 values less than ('PROMO PLATED
BRASS')
,
partition ptp91 values less than ('PROMO PLATED
COPPER')
,
```

```
partition ptp92 values less than ('PROMO PLATED
NICKEL')
,
partition ptp93 values less than ('PROMO PLATED
STEEL')
,
partition ptp94 values less than ('PROMO PLATED
TIN')
,
partition ptp95 values less than ('PROMO POLISHED
BRASS')
,
partition ptp96 values less than ('PROMO POLISHED
COPPER')
,
partition ptp97 values less than ('PROMO POLISHED
NICKEL')
,
partition ptp98 values less than ('PROMO POLISHED
STEEL')
,
partition ptp99 values less than ('PROMO POLISHED
TIN')
,
partition ptp100 values less than ('SMALL
ANODIZED BRASS')
,
partition ptp101 values less than ('SMALL
ANODIZED COPPER')
,
partition ptp102 values less than ('SMALL
ANODIZED NICKEL')
,
partition ptp103 values less than ('SMALL
ANODIZED STEEL')
,
partition ptp104 values less than ('SMALL
ANODIZED TIN')
,
partition ptp105 values less than ('SMALL BRUSHED
BRASS')
,
partition ptp106 values less than ('SMALL BRUSHED
COPPER')
,
partition ptp107 values less than ('SMALL BRUSHED
NICKEL')
,
partition ptp108 values less than ('SMALL BRUSHED
STEEL')
,
partition ptp109 values less than ('SMALL BRUSHED
TIN')
,
partition ptp110 values less than ('SMALL
BURNISHED BRASS')
,
partition ptp111 values less than ('SMALL
BURNISHED COPPER')
,
partition ptp112 values less than ('SMALL
BURNISHED NICKEL')
,
partition ptp113 values less than ('SMALL
BURNISHED STEEL')
,
partition ptp114 values less than ('SMALL
BURNISHED TIN')
,
partition ptp115 values less than ('SMALL PLATED
BRASS')
,
partition ptp116 values less than ('SMALL PLATED
COPPER')
,
partition ptp117 values less than ('SMALL PLATED
NICKEL')
,
partition ptp118 values less than ('SMALL PLATED
STEEL')
,
partition ptp119 values less than ('SMALL PLATED
TIN')
,
partition ptp120 values less than ('SMALL
POLISHED BRASS')
,
partition ptp121 values less than ('SMALL
POLISHED COPPER')
,
partition ptp122 values less than ('SMALL
POLISHED NICKEL')
,
partition ptp123 values less than ('SMALL
POLISHED STEEL')
,
partition ptp124 values less than ('SMALL
POLISHED TIN')
,
partition ptp125 values less than ('STANDARD
ANODIZED BRASS')
,
partition ptp126 values less than ('STANDARD
ANODIZED COPPER')
,
partition ptp127 values less than ('STANDARD
ANODIZED NICKEL')
,
partition ptp128 values less than ('STANDARD
ANODIZED STEEL')
,
partition ptp129 values less than ('STANDARD
ANODIZED TIN')
,
partition ptp130 values less than ('STANDARD
BRUSHED BRASS')
,
partition ptp131 values less than ('STANDARD
BRUSHED COPPER')
,
partition ptp132 values less than ('STANDARD
BRUSHED NICKEL')
,
partition ptp133 values less than ('STANDARD
BRUSHED STEEL')
,
partition ptp134 values less than ('STANDARD
BRUSHED TIN')
,
partition ptp135 values less than ('STANDARD
BURNISHED BRASS')
,
partition ptp136 values less than ('STANDARD
BURNISHED COPPER')
,
partition ptp137 values less than ('STANDARD
BURNISHED NICKEL')
,
```

```
partition ptp138 values less than ('STANDARD
BURNISHED STEEL')
,
partition ptp139 values less than ('STANDARD
BURNISHED TIN')
,
partition ptp140 values less than ('STANDARD
PLATED BRASS')
,
partition ptp141 values less than ('STANDARD
PLATED COPPER')
,
partition ptp142 values less than ('STANDARD
PLATED NICKEL')
,
partition ptp143 values less than ('STANDARD
PLATED STEEL')
,
partition ptp144 values less than ('STANDARD
PLATED TIN')
,
partition ptp145 values less than ('STANDARD
POLISHED BRASS')
,
partition ptp146 values less than ('STANDARD
POLISHED COPPER')
,
partition ptp147 values less than ('STANDARD
POLISHED NICKEL')
,
partition ptp148 values less than ('STANDARD
POLISHED STEEL')
,
partition ptp149 values less than ('STANDARD
POLISHED TIN')
,
partition ptp150 values less than (MAXVALUE)
)
;
}
*time=Begin creating index p_cbp
*sql
{
connect tpcd/tpcd;
create index p_cbp
on parts (p_container,p_brand,p_partkey)
pctfree 1
nologging
compute statistics
tablespace ts_mj_ppssrn
storage (initial 2m next 2m maxextents unlimited
pctincrease 0)
parallel (degree 20)
global partition by range (p_container,p_brand)
(
partition pcbp1 values less than ('JUMBO
BAG','Brand#111')
,
partition pcbp2 values less than ('JUMBO
BAG','Brand#121')
,
partition pcbp3 values less than ('JUMBO
BAG','Brand#131')
,
partition pcbp4 values less than ('JUMBO
BAG','Brand#141')
,
partition pcbp5 values less than ('JUMBO
BAG','Brand#151')
,
partition pcbp6 values less than ('JUMBO
BAG','Brand#211')
,
partition pcbp7 values less than ('JUMBO
BAG','Brand#221')
,
partition pcbp8 values less than ('JUMBO
BAG','Brand#231')
,
partition pcbp9 values less than ('JUMBO
BAG','Brand#241')
,
partition pcbp10 values less than ('JUMBO
BAG','Brand#251')
,
partition pcbp11 values less than ('JUMBO
BAG','Brand#311')
,
partition pcbp12 values less than ('JUMBO
BAG','Brand#321')
,
partition pcbp13 values less than ('JUMBO
BAG','Brand#331')
,
partition pcbp14 values less than ('JUMBO
BAG','Brand#341')
,
partition pcbp15 values less than ('JUMBO
BAG','Brand#351')
,
partition pcbp16 values less than ('JUMBO
BAG','Brand#411')
,
partition pcbp17 values less than ('JUMBO
BAG','Brand#421')
,
partition pcbp18 values less than ('JUMBO
BAG','Brand#431')
,
partition pcbp19 values less than ('JUMBO
BAG','Brand#441')
,
partition pcbp20 values less than ('JUMBO
BAG','Brand#451')
,
partition pcbp21 values less than ('JUMBO
BAG','Brand#511')
,
partition pcbp22 values less than ('JUMBO
BAG','Brand#521')
,
partition pcbp23 values less than ('JUMBO
BAG','Brand#531')
,
partition pcbp24 values less than ('JUMBO
BAG','Brand#541')
,
partition pcbp25 values less than ('JUMBO
BAG','Brand#551')
,
partition pcbp26 values less than ('JUMBO
BOX','Brand#111')
,
partition pcbp27 values less than ('JUMBO
BOX','Brand#121')
,
partition pcbp28 values less than ('JUMBO
BOX','Brand#131')
```

```
,
partition pcbp29 values less than ('JUMBO
BOX','Brand#141')
,
partition pcbp30 values less than ('JUMBO
BOX','Brand#151')
,
partition pcbp31 values less than ('JUMBO
BOX','Brand#211')
,
partition pcbp32 values less than ('JUMBO
BOX','Brand#221')
,
partition pcbp33 values less than ('JUMBO
BOX','Brand#231')
,
partition pcbp34 values less than ('JUMBO
BOX','Brand#241')
,
partition pcbp35 values less than ('JUMBO
BOX','Brand#251')
,
partition pcbp36 values less than ('JUMBO
BOX','Brand#311')
,
partition pcbp37 values less than ('JUMBO
BOX','Brand#321')
,
partition pcbp38 values less than ('JUMBO
BOX','Brand#331')
,
partition pcbp39 values less than ('JUMBO
BOX','Brand#341')
,
partition pcbp40 values less than ('JUMBO
BOX','Brand#351')
,
partition pcbp41 values less than ('JUMBO
BOX','Brand#411')
,
partition pcbp42 values less than ('JUMBO
BOX','Brand#421')
,
partition pcbp43 values less than ('JUMBO
BOX','Brand#431')
,
partition pcbp44 values less than ('JUMBO
BOX','Brand#441')
,
partition pcbp45 values less than ('JUMBO
BOX','Brand#451')
,
partition pcbp46 values less than ('JUMBO
BOX','Brand#511')
,
partition pcbp47 values less than ('JUMBO
BOX','Brand#521')
,
partition pcbp48 values less than ('JUMBO
BOX','Brand#531')
,
partition pcbp49 values less than ('JUMBO
BOX','Brand#541')
,
partition pcbp50 values less than ('JUMBO
BOX','Brand#551')
,
partition pcbp51 values less than ('JUMBO
CAN','Brand#111')
,
partition pcbp52 values less than ('JUMBO
CAN','Brand#121')
,
partition pcbp53 values less than ('JUMBO
CAN','Brand#131')
,
partition pcbp54 values less than ('JUMBO
CAN','Brand#141')
,
partition pcbp55 values less than ('JUMBO
CAN','Brand#151')
,
partition pcbp56 values less than ('JUMBO
CAN','Brand#211')
,
partition pcbp57 values less than ('JUMBO
CAN','Brand#221')
,
partition pcbp58 values less than ('JUMBO
CAN','Brand#231')
,
partition pcbp59 values less than ('JUMBO
CAN','Brand#241')
,
partition pcbp60 values less than ('JUMBO
CAN','Brand#251')
,
partition pcbp61 values less than ('JUMBO
CAN','Brand#311')
,
partition pcbp62 values less than ('JUMBO
CAN','Brand#321')
,
partition pcbp63 values less than ('JUMBO
CAN','Brand#331')
,
partition pcbp64 values less than ('JUMBO
CAN','Brand#341')
,
partition pcbp65 values less than ('JUMBO
CAN','Brand#351')
,
partition pcbp66 values less than ('JUMBO
CAN','Brand#411')
,
partition pcbp67 values less than ('JUMBO
CAN','Brand#421')
,
partition pcbp68 values less than ('JUMBO
CAN','Brand#431')
,
partition pcbp69 values less than ('JUMBO
CAN','Brand#441')
,
partition pcbp70 values less than ('JUMBO
CAN','Brand#451')
,
partition pcbp71 values less than ('JUMBO
CAN','Brand#511')
,
partition pcbp72 values less than ('JUMBO
CAN','Brand#521')
,
partition pcbp73 values less than ('JUMBO
CAN','Brand#531')
,
partition pcbp74 values less than ('JUMBO
CAN','Brand#541')
```

```
,
partition pcbp75 values less than ('JUMBO
CAN','Brand#551')
,
partition pcbp76 values less than ('JUMBO
CASE','Brand#111')
,
partition pcbp77 values less than ('JUMBO
CASE','Brand#121')
,
partition pcbp78 values less than ('JUMBO
CASE','Brand#131')
,
partition pcbp79 values less than ('JUMBO
CASE','Brand#141')
,
partition pcbp80 values less than ('JUMBO
CASE','Brand#151')
,
partition pcbp81 values less than ('JUMBO
CASE','Brand#211')
,
partition pcbp82 values less than ('JUMBO
CASE','Brand#221')
,
partition pcbp83 values less than ('JUMBO
CASE','Brand#231')
,
partition pcbp84 values less than ('JUMBO
CASE','Brand#241')
,
partition pcbp85 values less than ('JUMBO
CASE','Brand#251')
,
partition pcbp86 values less than ('JUMBO
CASE','Brand#311')
,
partition pcbp87 values less than ('JUMBO
CASE','Brand#321')
,
partition pcbp88 values less than ('JUMBO
CASE','Brand#331')
,
partition pcbp89 values less than ('JUMBO
CASE','Brand#341')
,
partition pcbp90 values less than ('JUMBO
CASE','Brand#351')
,
partition pcbp91 values less than ('JUMBO
CASE','Brand#411')
,
partition pcbp92 values less than ('JUMBO
CASE','Brand#421')
,
partition pcbp93 values less than ('JUMBO
CASE','Brand#431')
,
partition pcbp94 values less than ('JUMBO
CASE','Brand#441')
,
partition pcbp95 values less than ('JUMBO
CASE','Brand#451')
,
partition pcbp96 values less than ('JUMBO
CASE','Brand#511')
,
partition pcbp97 values less than ('JUMBO
CASE','Brand#521')
,
partition pcbp98 values less than ('JUMBO
CASE','Brand#531')
,
partition pcbp99 values less than ('JUMBO
CASE','Brand#541')
,
partition pcbp100 values less than ('JUMBO
CASE','Brand#551')
,
partition pcbp101 values less than ('JUMBO
DRUM','Brand#111')
,
partition pcbp102 values less than ('JUMBO
DRUM','Brand#121')
,
partition pcbp103 values less than ('JUMBO
DRUM','Brand#131')
,
partition pcbp104 values less than ('JUMBO
DRUM','Brand#141')
,
partition pcbp105 values less than ('JUMBO
DRUM','Brand#151')
,
partition pcbp106 values less than ('JUMBO
DRUM','Brand#211')
,
partition pcbp107 values less than ('JUMBO
DRUM','Brand#221')
,
partition pcbp108 values less than ('JUMBO
DRUM','Brand#231')
,
partition pcbp109 values less than ('JUMBO
DRUM','Brand#241')
,
partition pcbp110 values less than ('JUMBO
DRUM','Brand#251')
,
partition pcbp111 values less than ('JUMBO
DRUM','Brand#311')
,
partition pcbp112 values less than ('JUMBO
DRUM','Brand#321')
,
partition pcbp113 values less than ('JUMBO
DRUM','Brand#331')
,
partition pcbp114 values less than ('JUMBO
DRUM','Brand#341')
,
partition pcbp115 values less than ('JUMBO
DRUM','Brand#351')
,
partition pcbp116 values less than ('JUMBO
DRUM','Brand#411')
,
partition pcbp117 values less than ('JUMBO
DRUM','Brand#421')
,
partition pcbp118 values less than ('JUMBO
DRUM','Brand#431')
,
partition pcbp119 values less than ('JUMBO
DRUM','Brand#441')
,
partition pcbp120 values less than ('JUMBO
DRUM','Brand#451')
```

```
,
partition pcbp121 values less than ('JUMBO
DRUM','Brand#511')
,
partition pcbp122 values less than ('JUMBO
DRUM','Brand#521')
,
partition pcbp123 values less than ('JUMBO
DRUM','Brand#531')
,
partition pcbp124 values less than ('JUMBO
DRUM','Brand#541')
,
partition pcbp125 values less than ('JUMBO
DRUM','Brand#551')
,
partition pcbp126 values less than ('JUMBO
JAR','Brand#111')
,
partition pcbp127 values less than ('JUMBO
JAR','Brand#121')
,
partition pcbp128 values less than ('JUMBO
JAR','Brand#131')
,
partition pcbp129 values less than ('JUMBO
JAR','Brand#141')
,
partition pcbp130 values less than ('JUMBO
JAR','Brand#151')
,
partition pcbp131 values less than ('JUMBO
JAR','Brand#211')
,
partition pcbp132 values less than ('JUMBO
JAR','Brand#221')
,
partition pcbp133 values less than ('JUMBO
JAR','Brand#231')
,
partition pcbp134 values less than ('JUMBO
JAR','Brand#241')
,
partition pcbp135 values less than ('JUMBO
JAR','Brand#251')
,
partition pcbp136 values less than ('JUMBO
JAR','Brand#311')
,
partition pcbp137 values less than ('JUMBO
JAR','Brand#321')
,
partition pcbp138 values less than ('JUMBO
JAR','Brand#331')
,
partition pcbp139 values less than ('JUMBO
JAR','Brand#341')
,
partition pcbp140 values less than ('JUMBO
JAR','Brand#351')
,
partition pcbp141 values less than ('JUMBO
JAR','Brand#411')
,
partition pcbp142 values less than ('JUMBO
JAR','Brand#421')
,
partition pcbp143 values less than ('JUMBO
JAR','Brand#431')
,
partition pcbp144 values less than ('JUMBO
JAR','Brand#441')
,
partition pcbp145 values less than ('JUMBO
JAR','Brand#451')
,
partition pcbp146 values less than ('JUMBO
JAR','Brand#511')
,
partition pcbp147 values less than ('JUMBO
JAR','Brand#521')
,
partition pcbp148 values less than ('JUMBO
JAR','Brand#531')
,
partition pcbp149 values less than ('JUMBO
JAR','Brand#541')
,
partition pcbp150 values less than ('JUMBO
JAR','Brand#551')
,
partition pcbp151 values less than ('JUMBO
PACK','Brand#111')
,
partition pcbp152 values less than ('JUMBO
PACK','Brand#121')
,
partition pcbp153 values less than ('JUMBO
PACK','Brand#131')
,
partition pcbp154 values less than ('JUMBO
PACK','Brand#141')
,
partition pcbp155 values less than ('JUMBO
PACK','Brand#151')
,
partition pcbp156 values less than ('JUMBO
PACK','Brand#211')
,
partition pcbp157 values less than ('JUMBO
PACK','Brand#221')
,
partition pcbp158 values less than ('JUMBO
PACK','Brand#231')
,
partition pcbp159 values less than ('JUMBO
PACK','Brand#241')
,
partition pcbp160 values less than ('JUMBO
PACK','Brand#251')
,
partition pcbp161 values less than ('JUMBO
PACK','Brand#311')
,
partition pcbp162 values less than ('JUMBO
PACK','Brand#321')
,
partition pcbp163 values less than ('JUMBO
PACK','Brand#331')
,
partition pcbp164 values less than ('JUMBO
PACK','Brand#341')
,
partition pcbp165 values less than ('JUMBO
PACK','Brand#351')
,
partition pcbp166 values less than ('JUMBO
PACK','Brand#411')
```

```
,
partition pcbp167 values less than ('JUMBO
PACK','Brand#421')
,
partition pcbp168 values less than ('JUMBO
PACK','Brand#431')
,
partition pcbp169 values less than ('JUMBO
PACK','Brand#441')
,
partition pcbp170 values less than ('JUMBO
PACK','Brand#451')
,
partition pcbp171 values less than ('JUMBO
PACK','Brand#511')
,
partition pcbp172 values less than ('JUMBO
PACK','Brand#521')
,
partition pcbp173 values less than ('JUMBO
PACK','Brand#531')
,
partition pcbp174 values less than ('JUMBO
PACK','Brand#541')
,
partition pcbp175 values less than ('JUMBO
PACK','Brand#551')
,
partition pcbp176 values less than ('JUMBO
PKG','Brand#111')
,
partition pcbp177 values less than ('JUMBO
PKG','Brand#121')
,
partition pcbp178 values less than ('JUMBO
PKG','Brand#131')
,
partition pcbp179 values less than ('JUMBO
PKG','Brand#141')
,
partition pcbp180 values less than ('JUMBO
PKG','Brand#151')
,
partition pcbp181 values less than ('JUMBO
PKG','Brand#211')
,
partition pcbp182 values less than ('JUMBO
PKG','Brand#221')
,
partition pcbp183 values less than ('JUMBO
PKG','Brand#231')
,
partition pcbp184 values less than ('JUMBO
PKG','Brand#241')
,
partition pcbp185 values less than ('JUMBO
PKG','Brand#251')
,
partition pcbp186 values less than ('JUMBO
PKG','Brand#311')
,
partition pcbp187 values less than ('JUMBO
PKG','Brand#321')
,
partition pcbp188 values less than ('JUMBO
PKG','Brand#331')
,
partition pcbp189 values less than ('JUMBO
PKG','Brand#341')
,
partition pcbp190 values less than ('JUMBO
PKG','Brand#351')
,
partition pcbp191 values less than ('JUMBO
PKG','Brand#411')
,
partition pcbp192 values less than ('JUMBO
PKG','Brand#421')
,
partition pcbp193 values less than ('JUMBO
PKG','Brand#431')
,
partition pcbp194 values less than ('JUMBO
PKG','Brand#441')
,
partition pcbp195 values less than ('JUMBO
PKG','Brand#451')
,
partition pcbp196 values less than ('JUMBO
PKG','Brand#511')
,
partition pcbp197 values less than ('JUMBO
PKG','Brand#521')
,
partition pcbp198 values less than ('JUMBO
PKG','Brand#531')
,
partition pcbp199 values less than ('JUMBO
PKG','Brand#541')
,
partition pcbp200 values less than ('JUMBO
PKG','Brand#551')
,
partition pcbp201 values less than ('LG
BAG','Brand#111')
,
partition pcbp202 values less than ('LG
BAG','Brand#121')
,
partition pcbp203 values less than ('LG
BAG','Brand#131')
,
partition pcbp204 values less than ('LG
BAG','Brand#141')
,
partition pcbp205 values less than ('LG
BAG','Brand#151')
,
partition pcbp206 values less than ('LG
BAG','Brand#211')
,
partition pcbp207 values less than ('LG
BAG','Brand#221')
,
partition pcbp208 values less than ('LG
BAG','Brand#231')
,
partition pcbp209 values less than ('LG
BAG','Brand#241')
,
partition pcbp210 values less than ('LG
BAG','Brand#251')
,
partition pcbp211 values less than ('LG
BAG','Brand#311')
,
partition pcbp212 values less than ('LG
BAG','Brand#321')
```

```
,
partition pcbp213 values less than ('LG
BAG','Brand#331')
,
partition pcbp214 values less than ('LG
BAG','Brand#341')
,
partition pcbp215 values less than ('LG
BAG','Brand#351')
,
partition pcbp216 values less than ('LG
BAG','Brand#411')
,
partition pcbp217 values less than ('LG
BAG','Brand#421')
,
partition pcbp218 values less than ('LG
BAG','Brand#431')
,
partition pcbp219 values less than ('LG
BAG','Brand#441')
,
partition pcbp220 values less than ('LG
BAG','Brand#451')
,
partition pcbp221 values less than ('LG
BAG','Brand#511')
,
partition pcbp222 values less than ('LG
BAG','Brand#521')
,
partition pcbp223 values less than ('LG
BAG','Brand#531')
,
partition pcbp224 values less than ('LG
BAG','Brand#541')
,
partition pcbp225 values less than ('LG
BAG','Brand#551')
,
partition pcbp226 values less than ('LG
BOX','Brand#111')
,
partition pcbp227 values less than ('LG
BOX','Brand#121')
,
partition pcbp228 values less than ('LG
BOX','Brand#131')
,
partition pcbp229 values less than ('LG
BOX','Brand#141')
,
partition pcbp230 values less than ('LG
BOX','Brand#151')
,
partition pcbp231 values less than ('LG
BOX','Brand#211')
,
partition pcbp232 values less than ('LG
BOX','Brand#221')
,
partition pcbp233 values less than ('LG
BOX','Brand#231')
,
partition pcbp234 values less than ('LG
BOX','Brand#241')
,
partition pcbp235 values less than ('LG
BOX','Brand#251')
,
partition pcbp236 values less than ('LG
BOX','Brand#311')
,
partition pcbp237 values less than ('LG
BOX','Brand#321')
,
partition pcbp238 values less than ('LG
BOX','Brand#331')
,
partition pcbp239 values less than ('LG
BOX','Brand#341')
,
partition pcbp240 values less than ('LG
BOX','Brand#351')
,
partition pcbp241 values less than ('LG
BOX','Brand#411')
,
partition pcbp242 values less than ('LG
BOX','Brand#421')
,
partition pcbp243 values less than ('LG
BOX','Brand#431')
,
partition pcbp244 values less than ('LG
BOX','Brand#441')
,
partition pcbp245 values less than ('LG
BOX','Brand#451')
,
partition pcbp246 values less than ('LG
BOX','Brand#511')
,
partition pcbp247 values less than ('LG
BOX','Brand#521')
,
partition pcbp248 values less than ('LG
BOX','Brand#531')
,
partition pcbp249 values less than ('LG
BOX','Brand#541')
,
partition pcbp250 values less than ('LG
BOX','Brand#551')
,
partition pcbp251 values less than ('LG
CAN','Brand#111')
,
partition pcbp252 values less than ('LG
CAN','Brand#121')
,
partition pcbp253 values less than ('LG
CAN','Brand#131')
,
partition pcbp254 values less than ('LG
CAN','Brand#141')
,
partition pcbp255 values less than ('LG
CAN','Brand#151')
,
partition pcbp256 values less than ('LG
CAN','Brand#211')
,
partition pcbp257 values less than ('LG
CAN','Brand#221')
,
partition pcbp258 values less than ('LG
CAN','Brand#231')
```

```
,
partition pcbp259 values less than ('LG
CAN','Brand#241')
,
partition pcbp260 values less than ('LG
CAN','Brand#251')
,
partition pcbp261 values less than ('LG
CAN','Brand#311')
,
partition pcbp262 values less than ('LG
CAN','Brand#321')
,
partition pcbp263 values less than ('LG
CAN','Brand#331')
,
partition pcbp264 values less than ('LG
CAN','Brand#341')
,
partition pcbp265 values less than ('LG
CAN','Brand#351')
,
partition pcbp266 values less than ('LG
CAN','Brand#411')
,
partition pcbp267 values less than ('LG
CAN','Brand#421')
,
partition pcbp268 values less than ('LG
CAN','Brand#431')
,
partition pcbp269 values less than ('LG
CAN','Brand#441')
,
partition pcbp270 values less than ('LG
CAN','Brand#451')
,
partition pcbp271 values less than ('LG
CAN','Brand#511')
,
partition pcbp272 values less than ('LG
CAN','Brand#521')
,
partition pcbp273 values less than ('LG
CAN','Brand#531')
,
partition pcbp274 values less than ('LG
CAN','Brand#541')
,
partition pcbp275 values less than ('LG
CAN','Brand#551')
,
partition pcbp276 values less than ('LG
CASE','Brand#111')
,
partition pcbp277 values less than ('LG
CASE','Brand#121')
,
partition pcbp278 values less than ('LG
CASE','Brand#131')
,
partition pcbp279 values less than ('LG
CASE','Brand#141')
,
partition pcbp280 values less than ('LG
CASE','Brand#151')
,
partition pcbp281 values less than ('LG
CASE','Brand#211')
,
partition pcbp282 values less than ('LG
CASE','Brand#221')
,
partition pcbp283 values less than ('LG
CASE','Brand#231')
,
partition pcbp284 values less than ('LG
CASE','Brand#241')
,
partition pcbp285 values less than ('LG
CASE','Brand#251')
,
partition pcbp286 values less than ('LG
CASE','Brand#311')
,
partition pcbp287 values less than ('LG
CASE','Brand#321')
,
partition pcbp288 values less than ('LG
CASE','Brand#331')
,
partition pcbp289 values less than ('LG
CASE','Brand#341')
,
partition pcbp290 values less than ('LG
CASE','Brand#351')
,
partition pcbp291 values less than ('LG
CASE','Brand#411')
,
partition pcbp292 values less than ('LG
CASE','Brand#421')
,
partition pcbp293 values less than ('LG
CASE','Brand#431')
,
partition pcbp294 values less than ('LG
CASE','Brand#441')
,
partition pcbp295 values less than ('LG
CASE','Brand#451')
,
partition pcbp296 values less than ('LG
CASE','Brand#511')
,
partition pcbp297 values less than ('LG
CASE','Brand#521')
,
partition pcbp298 values less than ('LG
CASE','Brand#531')
,
partition pcbp299 values less than ('LG
CASE','Brand#541')
,
partition pcbp300 values less than ('LG
CASE','Brand#551')
,
partition pcbp301 values less than ('LG
DRUM','Brand#111')
,
partition pcbp302 values less than ('LG
DRUM','Brand#121')
,
partition pcbp303 values less than ('LG
DRUM','Brand#131')
,
partition pcbp304 values less than ('LG
DRUM','Brand#141')
```

```
,
partition pcbp305 values less than ('LG
DRUM','Brand#151')
,
partition pcbp306 values less than ('LG
DRUM','Brand#211')
,
partition pcbp307 values less than ('LG
DRUM','Brand#221')
,
partition pcbp308 values less than ('LG
DRUM','Brand#231')
,
partition pcbp309 values less than ('LG
DRUM','Brand#241')
,
partition pcbp310 values less than ('LG
DRUM','Brand#251')
,
partition pcbp311 values less than ('LG
DRUM','Brand#311')
,
partition pcbp312 values less than ('LG
DRUM','Brand#321')
,
partition pcbp313 values less than ('LG
DRUM','Brand#331')
,
partition pcbp314 values less than ('LG
DRUM','Brand#341')
,
partition pcbp315 values less than ('LG
DRUM','Brand#351')
,
partition pcbp316 values less than ('LG
DRUM','Brand#411')
,
partition pcbp317 values less than ('LG
DRUM','Brand#421')
,
partition pcbp318 values less than ('LG
DRUM','Brand#431')
,
partition pcbp319 values less than ('LG
DRUM','Brand#441')
,
partition pcbp320 values less than ('LG
DRUM','Brand#451')
,
partition pcbp321 values less than ('LG
DRUM','Brand#511')
,
partition pcbp322 values less than ('LG
DRUM','Brand#521')
,
partition pcbp323 values less than ('LG
DRUM','Brand#531')
,
partition pcbp324 values less than ('LG
DRUM','Brand#541')
,
partition pcbp325 values less than ('LG
DRUM','Brand#551')
,
partition pcbp326 values less than ('LG
JAR','Brand#111')
,
partition pcbp327 values less than ('LG
JAR','Brand#121')
,
partition pcbp328 values less than ('LG
JAR','Brand#131')
,
partition pcbp329 values less than ('LG
JAR','Brand#141')
,
partition pcbp330 values less than ('LG
JAR','Brand#151')
,
partition pcbp331 values less than ('LG
JAR','Brand#211')
,
partition pcbp332 values less than ('LG
JAR','Brand#221')
,
partition pcbp333 values less than ('LG
JAR','Brand#231')
,
partition pcbp334 values less than ('LG
JAR','Brand#241')
,
partition pcbp335 values less than ('LG
JAR','Brand#251')
,
partition pcbp336 values less than ('LG
JAR','Brand#311')
,
partition pcbp337 values less than ('LG
JAR','Brand#321')
,
partition pcbp338 values less than ('LG
JAR','Brand#331')
,
partition pcbp339 values less than ('LG
JAR','Brand#341')
,
partition pcbp340 values less than ('LG
JAR','Brand#351')
,
partition pcbp341 values less than ('LG
JAR','Brand#411')
,
partition pcbp342 values less than ('LG
JAR','Brand#421')
,
partition pcbp343 values less than ('LG
JAR','Brand#431')
,
partition pcbp344 values less than ('LG
JAR','Brand#441')
,
partition pcbp345 values less than ('LG
JAR','Brand#451')
,
partition pcbp346 values less than ('LG
JAR','Brand#511')
,
partition pcbp347 values less than ('LG
JAR','Brand#521')
,
partition pcbp348 values less than ('LG
JAR','Brand#531')
,
partition pcbp349 values less than ('LG
JAR','Brand#541')
,
partition pcbp350 values less than ('LG
JAR','Brand#551')
```

```
,
partition pcbp351 values less than ('LG
PACK','Brand#111')
,
partition pcbp352 values less than ('LG
PACK','Brand#121')
,
partition pcbp353 values less than ('LG
PACK','Brand#131')
,
partition pcbp354 values less than ('LG
PACK','Brand#141')
,
partition pcbp355 values less than ('LG
PACK','Brand#151')
,
partition pcbp356 values less than ('LG
PACK','Brand#211')
,
partition pcbp357 values less than ('LG
PACK','Brand#221')
,
partition pcbp358 values less than ('LG
PACK','Brand#231')
,
partition pcbp359 values less than ('LG
PACK','Brand#241')
,
partition pcbp360 values less than ('LG
PACK','Brand#251')
,
partition pcbp361 values less than ('LG
PACK','Brand#311')
,
partition pcbp362 values less than ('LG
PACK','Brand#321')
,
partition pcbp363 values less than ('LG
PACK','Brand#331')
,
partition pcbp364 values less than ('LG
PACK','Brand#341')
,
partition pcbp365 values less than ('LG
PACK','Brand#351')
,
partition pcbp366 values less than ('LG
PACK','Brand#411')
,
partition pcbp367 values less than ('LG
PACK','Brand#421')
,
partition pcbp368 values less than ('LG
PACK','Brand#431')
,
partition pcbp369 values less than ('LG
PACK','Brand#441')
,
partition pcbp370 values less than ('LG
PACK','Brand#451')
,
partition pcbp371 values less than ('LG
PACK','Brand#511')
,
partition pcbp372 values less than ('LG
PACK','Brand#521')
,
partition pcbp373 values less than ('LG
PACK','Brand#531')
,
partition pcbp374 values less than ('LG
PACK','Brand#541')
,
partition pcbp375 values less than ('LG
PACK','Brand#551')
,
partition pcbp376 values less than ('LG
PKG','Brand#111')
,
partition pcbp377 values less than ('LG
PKG','Brand#121')
,
partition pcbp378 values less than ('LG
PKG','Brand#131')
,
partition pcbp379 values less than ('LG
PKG','Brand#141')
,
partition pcbp380 values less than ('LG
PKG','Brand#151')
,
partition pcbp381 values less than ('LG
PKG','Brand#211')
,
partition pcbp382 values less than ('LG
PKG','Brand#221')
,
partition pcbp383 values less than ('LG
PKG','Brand#231')
,
partition pcbp384 values less than ('LG
PKG','Brand#241')
,
partition pcbp385 values less than ('LG
PKG','Brand#251')
,
partition pcbp386 values less than ('LG
PKG','Brand#311')
,
partition pcbp387 values less than ('LG
PKG','Brand#321')
,
partition pcbp388 values less than ('LG
PKG','Brand#331')
,
partition pcbp389 values less than ('LG
PKG','Brand#341')
,
partition pcbp390 values less than ('LG
PKG','Brand#351')
,
partition pcbp391 values less than ('LG
PKG','Brand#411')
,
partition pcbp392 values less than ('LG
PKG','Brand#421')
,
partition pcbp393 values less than ('LG
PKG','Brand#431')
,
partition pcbp394 values less than ('LG
PKG','Brand#441')
,
partition pcbp395 values less than ('LG
PKG','Brand#451')
,
partition pcbp396 values less than ('LG
PKG','Brand#511')
```

```
,
partition pcbp397 values less than ('LG
PKG','Brand#521')
,
partition pcbp398 values less than ('LG
PKG','Brand#531')
,
partition pcbp399 values less than ('LG
PKG','Brand#541')
,
partition pcbp400 values less than ('LG
PKG','Brand#551')
,
partition pcbp401 values less than ('MED
BAG','Brand#111')
,
partition pcbp402 values less than ('MED
BAG','Brand#121')
,
partition pcbp403 values less than ('MED
BAG','Brand#131')
,
partition pcbp404 values less than ('MED
BAG','Brand#141')
,
partition pcbp405 values less than ('MED
BAG','Brand#151')
,
partition pcbp406 values less than ('MED
BAG','Brand#211')
,
partition pcbp407 values less than ('MED
BAG','Brand#221')
,
partition pcbp408 values less than ('MED
BAG','Brand#231')
,
partition pcbp409 values less than ('MED
BAG','Brand#241')
,
partition pcbp410 values less than ('MED
BAG','Brand#251')
,
partition pcbp411 values less than ('MED
BAG','Brand#311')
,
partition pcbp412 values less than ('MED
BAG','Brand#321')
,
partition pcbp413 values less than ('MED
BAG','Brand#331')
,
partition pcbp414 values less than ('MED
BAG','Brand#341')
,
partition pcbp415 values less than ('MED
BAG','Brand#351')
,
partition pcbp416 values less than ('MED
BAG','Brand#411')
,
partition pcbp417 values less than ('MED
BAG','Brand#421')
,
partition pcbp418 values less than ('MED
BAG','Brand#431')
,
partition pcbp419 values less than ('MED
BAG','Brand#441')
,
partition pcbp420 values less than ('MED
BAG','Brand#451')
,
partition pcbp421 values less than ('MED
BAG','Brand#511')
,
partition pcbp422 values less than ('MED
BAG','Brand#521')
,
partition pcbp423 values less than ('MED
BAG','Brand#531')
,
partition pcbp424 values less than ('MED
BAG','Brand#541')
,
partition pcbp425 values less than ('MED
BAG','Brand#551')
,
partition pcbp426 values less than ('MED
BOX','Brand#111')
,
partition pcbp427 values less than ('MED
BOX','Brand#121')
,
partition pcbp428 values less than ('MED
BOX','Brand#131')
,
partition pcbp429 values less than ('MED
BOX','Brand#141')
,
partition pcbp430 values less than ('MED
BOX','Brand#151')
,
partition pcbp431 values less than ('MED
BOX','Brand#211')
,
partition pcbp432 values less than ('MED
BOX','Brand#221')
,
partition pcbp433 values less than ('MED
BOX','Brand#231')
,
partition pcbp434 values less than ('MED
BOX','Brand#241')
,
partition pcbp435 values less than ('MED
BOX','Brand#251')
,
partition pcbp436 values less than ('MED
BOX','Brand#311')
,
partition pcbp437 values less than ('MED
BOX','Brand#321')
,
partition pcbp438 values less than ('MED
BOX','Brand#331')
,
partition pcbp439 values less than ('MED
BOX','Brand#341')
,
partition pcbp440 values less than ('MED
BOX','Brand#351')
,
partition pcbp441 values less than ('MED
BOX','Brand#411')
,
partition pcbp442 values less than ('MED
BOX','Brand#421')
```

```
,
partition pcbp443 values less than ('MED
BOX','Brand#431')
,
partition pcbp444 values less than ('MED
BOX','Brand#441')
,
partition pcbp445 values less than ('MED
BOX','Brand#451')
,
partition pcbp446 values less than ('MED
BOX','Brand#511')
,
partition pcbp447 values less than ('MED
BOX','Brand#521')
,
partition pcbp448 values less than ('MED
BOX','Brand#531')
,
partition pcbp449 values less than ('MED
BOX','Brand#541')
,
partition pcbp450 values less than ('MED
BOX','Brand#551')
,
partition pcbp451 values less than ('MED
CAN','Brand#111')
,
partition pcbp452 values less than ('MED
CAN','Brand#121')
,
partition pcbp453 values less than ('MED
CAN','Brand#131')
,
partition pcbp454 values less than ('MED
CAN','Brand#141')
,
partition pcbp455 values less than ('MED
CAN','Brand#151')
,
partition pcbp456 values less than ('MED
CAN','Brand#211')
,
partition pcbp457 values less than ('MED
CAN','Brand#221')
,
partition pcbp458 values less than ('MED
CAN','Brand#231')
,
partition pcbp459 values less than ('MED
CAN','Brand#241')
,
partition pcbp460 values less than ('MED
CAN','Brand#251')
,
partition pcbp461 values less than ('MED
CAN','Brand#311')
,
partition pcbp462 values less than ('MED
CAN','Brand#321')
,
partition pcbp463 values less than ('MED
CAN','Brand#331')
,
partition pcbp464 values less than ('MED
CAN','Brand#341')
,
partition pcbp465 values less than ('MED
CAN','Brand#351')
,
partition pcbp466 values less than ('MED
CAN','Brand#411')
,
partition pcbp467 values less than ('MED
CAN','Brand#421')
,
partition pcbp468 values less than ('MED
CAN','Brand#431')
,
partition pcbp469 values less than ('MED
CAN','Brand#441')
,
partition pcbp470 values less than ('MED
CAN','Brand#451')
,
partition pcbp471 values less than ('MED
CAN','Brand#511')
,
partition pcbp472 values less than ('MED
CAN','Brand#521')
,
partition pcbp473 values less than ('MED
CAN','Brand#531')
,
partition pcbp474 values less than ('MED
CAN','Brand#541')
,
partition pcbp475 values less than ('MED
CAN','Brand#551')
,
partition pcbp476 values less than ('MED
CASE','Brand#111')
,
partition pcbp477 values less than ('MED
CASE','Brand#121')
,
partition pcbp478 values less than ('MED
CASE','Brand#131')
,
partition pcbp479 values less than ('MED
CASE','Brand#141')
,
partition pcbp480 values less than ('MED
CASE','Brand#151')
,
partition pcbp481 values less than ('MED
CASE','Brand#211')
,
partition pcbp482 values less than ('MED
CASE','Brand#221')
,
partition pcbp483 values less than ('MED
CASE','Brand#231')
,
partition pcbp484 values less than ('MED
CASE','Brand#241')
,
partition pcbp485 values less than ('MED
CASE','Brand#251')
,
partition pcbp486 values less than ('MED
CASE','Brand#311')
,
partition pcbp487 values less than ('MED
CASE','Brand#321')
,
partition pcbp488 values less than ('MED
CASE','Brand#331')
```

```
,
partition pcbp489 values less than ('MED
CASE','Brand#341')
,
partition pcbp490 values less than ('MED
CASE','Brand#351')
,
partition pcbp491 values less than ('MED
CASE','Brand#411')
,
partition pcbp492 values less than ('MED
CASE','Brand#421')
,
partition pcbp493 values less than ('MED
CASE','Brand#431')
,
partition pcbp494 values less than ('MED
CASE','Brand#441')
,
partition pcbp495 values less than ('MED
CASE','Brand#451')
,
partition pcbp496 values less than ('MED
CASE','Brand#511')
,
partition pcbp497 values less than ('MED
CASE','Brand#521')
,
partition pcbp498 values less than ('MED
CASE','Brand#531')
,
partition pcbp499 values less than ('MED
CASE','Brand#541')
,
partition pcbp500 values less than ('MED
CASE','Brand#551')
,
partition pcbp501 values less than ('MED
DRUM','Brand#111')
,
partition pcbp502 values less than ('MED
DRUM','Brand#121')
,
partition pcbp503 values less than ('MED
DRUM','Brand#131')
,
partition pcbp504 values less than ('MED
DRUM','Brand#141')
,
partition pcbp505 values less than ('MED
DRUM','Brand#151')
,
partition pcbp506 values less than ('MED
DRUM','Brand#211')
,
partition pcbp507 values less than ('MED
DRUM','Brand#221')
,
partition pcbp508 values less than ('MED
DRUM','Brand#231')
,
partition pcbp509 values less than ('MED
DRUM','Brand#241')
,
partition pcbp510 values less than ('MED
DRUM','Brand#251')
,
partition pcbp511 values less than ('MED
DRUM','Brand#311')
,
partition pcbp512 values less than ('MED
DRUM','Brand#321')
,
partition pcbp513 values less than ('MED
DRUM','Brand#331')
,
partition pcbp514 values less than ('MED
DRUM','Brand#341')
,
partition pcbp515 values less than ('MED
DRUM','Brand#351')
,
partition pcbp516 values less than ('MED
DRUM','Brand#411')
,
partition pcbp517 values less than ('MED
DRUM','Brand#421')
,
partition pcbp518 values less than ('MED
DRUM','Brand#431')
,
partition pcbp519 values less than ('MED
DRUM','Brand#441')
,
partition pcbp520 values less than ('MED
DRUM','Brand#451')
,
partition pcbp521 values less than ('MED
DRUM','Brand#511')
,
partition pcbp522 values less than ('MED
DRUM','Brand#521')
,
partition pcbp523 values less than ('MED
DRUM','Brand#531')
,
partition pcbp524 values less than ('MED
DRUM','Brand#541')
,
partition pcbp525 values less than ('MED
DRUM','Brand#551')
,
partition pcbp526 values less than ('MED
JAR','Brand#111')
,
partition pcbp527 values less than ('MED
JAR','Brand#121')
,
partition pcbp528 values less than ('MED
JAR','Brand#131')
,
partition pcbp529 values less than ('MED
JAR','Brand#141')
,
partition pcbp530 values less than ('MED
JAR','Brand#151')
,
partition pcbp531 values less than ('MED
JAR','Brand#211')
,
partition pcbp532 values less than ('MED
JAR','Brand#221')
,
partition pcbp533 values less than ('MED
JAR','Brand#231')
,
partition pcbp534 values less than ('MED
JAR','Brand#241')
```

```
,
partition pcbp535 values less than ('MED
JAR','Brand#251')
,
partition pcbp536 values less than ('MED
JAR','Brand#311')
,
partition pcbp537 values less than ('MED
JAR','Brand#321')
,
partition pcbp538 values less than ('MED
JAR','Brand#331')
,
partition pcbp539 values less than ('MED
JAR','Brand#341')
,
partition pcbp540 values less than ('MED
JAR','Brand#351')
,
partition pcbp541 values less than ('MED
JAR','Brand#411')
,
partition pcbp542 values less than ('MED
JAR','Brand#421')
,
partition pcbp543 values less than ('MED
JAR','Brand#431')
,
partition pcbp544 values less than ('MED
JAR','Brand#441')
,
partition pcbp545 values less than ('MED
JAR','Brand#451')
,
partition pcbp546 values less than ('MED
JAR','Brand#511')
,
partition pcbp547 values less than ('MED
JAR','Brand#521')
,
partition pcbp548 values less than ('MED
JAR','Brand#531')
,
partition pcbp549 values less than ('MED
JAR','Brand#541')
,
partition pcbp550 values less than ('MED
JAR','Brand#551')
,
partition pcbp551 values less than ('MED
PACK','Brand#111')
,
partition pcbp552 values less than ('MED
PACK','Brand#121')
,
partition pcbp553 values less than ('MED
PACK','Brand#131')
,
partition pcbp554 values less than ('MED
PACK','Brand#141')
,
partition pcbp555 values less than ('MED
PACK','Brand#151')
,
partition pcbp556 values less than ('MED
PACK','Brand#211')
,
partition pcbp557 values less than ('MED
PACK','Brand#221')
,
partition pcbp558 values less than ('MED
PACK','Brand#231')
,
partition pcbp559 values less than ('MED
PACK','Brand#241')
,
partition pcbp560 values less than ('MED
PACK','Brand#251')
,
partition pcbp561 values less than ('MED
PACK','Brand#311')
,
partition pcbp562 values less than ('MED
PACK','Brand#321')
,
partition pcbp563 values less than ('MED
PACK','Brand#331')
,
partition pcbp564 values less than ('MED
PACK','Brand#341')
,
partition pcbp565 values less than ('MED
PACK','Brand#351')
,
partition pcbp566 values less than ('MED
PACK','Brand#411')
,
partition pcbp567 values less than ('MED
PACK','Brand#421')
,
partition pcbp568 values less than ('MED
PACK','Brand#431')
,
partition pcbp569 values less than ('MED
PACK','Brand#441')
,
partition pcbp570 values less than ('MED
PACK','Brand#451')
,
partition pcbp571 values less than ('MED
PACK','Brand#511')
,
partition pcbp572 values less than ('MED
PACK','Brand#521')
,
partition pcbp573 values less than ('MED
PACK','Brand#531')
,
partition pcbp574 values less than ('MED
PACK','Brand#541')
,
partition pcbp575 values less than ('MED
PACK','Brand#551')
,
partition pcbp576 values less than ('MED
PKG','Brand#111')
,
partition pcbp577 values less than ('MED
PKG','Brand#121')
,
partition pcbp578 values less than ('MED
PKG','Brand#131')
,
partition pcbp579 values less than ('MED
PKG','Brand#141')
,
partition pcbp580 values less than ('MED
PKG','Brand#151')
```

```
,
partition pcbp581 values less than ('MED
PKG','Brand#211')
,
partition pcbp582 values less than ('MED
PKG','Brand#221')
,
partition pcbp583 values less than ('MED
PKG','Brand#231')
,
partition pcbp584 values less than ('MED
PKG','Brand#241')
,
partition pcbp585 values less than ('MED
PKG','Brand#251')
,
partition pcbp586 values less than ('MED
PKG','Brand#311')
,
partition pcbp587 values less than ('MED
PKG','Brand#321')
,
partition pcbp588 values less than ('MED
PKG','Brand#331')
,
partition pcbp589 values less than ('MED
PKG','Brand#341')
,
partition pcbp590 values less than ('MED
PKG','Brand#351')
,
partition pcbp591 values less than ('MED
PKG','Brand#411')
,
partition pcbp592 values less than ('MED
PKG','Brand#421')
,
partition pcbp593 values less than ('MED
PKG','Brand#431')
,
partition pcbp594 values less than ('MED
PKG','Brand#441')
,
partition pcbp595 values less than ('MED
PKG','Brand#451')
,
partition pcbp596 values less than ('MED
PKG','Brand#511')
,
partition pcbp597 values less than ('MED
PKG','Brand#521')
,
partition pcbp598 values less than ('MED
PKG','Brand#531')
,
partition pcbp599 values less than ('MED
PKG','Brand#541')
,
partition pcbp600 values less than ('MED
PKG','Brand#551')
,
partition pcbp601 values less than ('SM
BAG','Brand#111')
,
partition pcbp602 values less than ('SM
BAG','Brand#121')
,
partition pcbp603 values less than ('SM
BAG','Brand#131')
,
partition pcbp604 values less than ('SM
BAG','Brand#141')
,
partition pcbp605 values less than ('SM
BAG','Brand#151')
,
partition pcbp606 values less than ('SM
BAG','Brand#211')
,
partition pcbp607 values less than ('SM
BAG','Brand#221')
,
partition pcbp608 values less than ('SM
BAG','Brand#231')
,
partition pcbp609 values less than ('SM
BAG','Brand#241')
,
partition pcbp610 values less than ('SM
BAG','Brand#251')
,
partition pcbp611 values less than ('SM
BAG','Brand#311')
,
partition pcbp612 values less than ('SM
BAG','Brand#321')
,
partition pcbp613 values less than ('SM
BAG','Brand#331')
,
partition pcbp614 values less than ('SM
BAG','Brand#341')
,
partition pcbp615 values less than ('SM
BAG','Brand#351')
,
partition pcbp616 values less than ('SM
BAG','Brand#411')
,
partition pcbp617 values less than ('SM
BAG','Brand#421')
,
partition pcbp618 values less than ('SM
BAG','Brand#431')
,
partition pcbp619 values less than ('SM
BAG','Brand#441')
,
partition pcbp620 values less than ('SM
BAG','Brand#451')
,
partition pcbp621 values less than ('SM
BAG','Brand#511')
,
partition pcbp622 values less than ('SM
BAG','Brand#521')
,
partition pcbp623 values less than ('SM
BAG','Brand#531')
,
partition pcbp624 values less than ('SM
BAG','Brand#541')
,
partition pcbp625 values less than ('SM
BAG','Brand#551')
,
partition pcbp626 values less than ('SM
BOX','Brand#111')
```

```
,
partition pcbp627 values less than ('SM
BOX','Brand#121')
,
partition pcbp628 values less than ('SM
BOX','Brand#131')
,
partition pcbp629 values less than ('SM
BOX','Brand#141')
,
partition pcbp630 values less than ('SM
BOX','Brand#151')
,
partition pcbp631 values less than ('SM
BOX','Brand#211')
,
partition pcbp632 values less than ('SM
BOX','Brand#221')
,
partition pcbp633 values less than ('SM
BOX','Brand#231')
,
partition pcbp634 values less than ('SM
BOX','Brand#241')
,
partition pcbp635 values less than ('SM
BOX','Brand#251')
,
partition pcbp636 values less than ('SM
BOX','Brand#311')
,
partition pcbp637 values less than ('SM
BOX','Brand#321')
,
partition pcbp638 values less than ('SM
BOX','Brand#331')
,
partition pcbp639 values less than ('SM
BOX','Brand#341')
,
partition pcbp640 values less than ('SM
BOX','Brand#351')
,
partition pcbp641 values less than ('SM
BOX','Brand#411')
,
partition pcbp642 values less than ('SM
BOX','Brand#421')
,
partition pcbp643 values less than ('SM
BOX','Brand#431')
,
partition pcbp644 values less than ('SM
BOX','Brand#441')
,
partition pcbp645 values less than ('SM
BOX','Brand#451')
,
partition pcbp646 values less than ('SM
BOX','Brand#511')
,
partition pcbp647 values less than ('SM
BOX','Brand#521')
,
partition pcbp648 values less than ('SM
BOX','Brand#531')
,
partition pcbp649 values less than ('SM
BOX','Brand#541')
,
partition pcbp650 values less than ('SM
BOX','Brand#551')
,
partition pcbp651 values less than ('SM
CAN','Brand#111')
,
partition pcbp652 values less than ('SM
CAN','Brand#121')
,
partition pcbp653 values less than ('SM
CAN','Brand#131')
,
partition pcbp654 values less than ('SM
CAN','Brand#141')
,
partition pcbp655 values less than ('SM
CAN','Brand#151')
,
partition pcbp656 values less than ('SM
CAN','Brand#211')
,
partition pcbp657 values less than ('SM
CAN','Brand#221')
,
partition pcbp658 values less than ('SM
CAN','Brand#231')
,
partition pcbp659 values less than ('SM
CAN','Brand#241')
,
partition pcbp660 values less than ('SM
CAN','Brand#251')
,
partition pcbp661 values less than ('SM
CAN','Brand#311')
,
partition pcbp662 values less than ('SM
CAN','Brand#321')
,
partition pcbp663 values less than ('SM
CAN','Brand#331')
,
partition pcbp664 values less than ('SM
CAN','Brand#341')
,
partition pcbp665 values less than ('SM
CAN','Brand#351')
,
partition pcbp666 values less than ('SM
CAN','Brand#411')
,
partition pcbp667 values less than ('SM
CAN','Brand#421')
,
partition pcbp668 values less than ('SM
CAN','Brand#431')
,
partition pcbp669 values less than ('SM
CAN','Brand#441')
,
partition pcbp670 values less than ('SM
CAN','Brand#451')
,
partition pcbp671 values less than ('SM
CAN','Brand#511')
,
partition pcbp672 values less than ('SM
CAN','Brand#521')
```

```
,
partition pcbp673 values less than ('SM
CAN','Brand#531')
,
partition pcbp674 values less than ('SM
CAN','Brand#541')
,
partition pcbp675 values less than ('SM
CAN','Brand#551')
,
partition pcbp676 values less than ('SM
CASE','Brand#111')
,
partition pcbp677 values less than ('SM
CASE','Brand#121')
,
partition pcbp678 values less than ('SM
CASE','Brand#131')
,
partition pcbp679 values less than ('SM
CASE','Brand#141')
,
partition pcbp680 values less than ('SM
CASE','Brand#151')
,
partition pcbp681 values less than ('SM
CASE','Brand#211')
,
partition pcbp682 values less than ('SM
CASE','Brand#221')
,
partition pcbp683 values less than ('SM
CASE','Brand#231')
,
partition pcbp684 values less than ('SM
CASE','Brand#241')
,
partition pcbp685 values less than ('SM
CASE','Brand#251')
,
partition pcbp686 values less than ('SM
CASE','Brand#311')
,
partition pcbp687 values less than ('SM
CASE','Brand#321')
,
partition pcbp688 values less than ('SM
CASE','Brand#331')
,
partition pcbp689 values less than ('SM
CASE','Brand#341')
,
partition pcbp690 values less than ('SM
CASE','Brand#351')
,
partition pcbp691 values less than ('SM
CASE','Brand#411')
,
partition pcbp692 values less than ('SM
CASE','Brand#421')
,
partition pcbp693 values less than ('SM
CASE','Brand#431')
,
partition pcbp694 values less than ('SM
CASE','Brand#441')
,
partition pcbp695 values less than ('SM
CASE','Brand#451')
,
partition pcbp696 values less than ('SM
CASE','Brand#511')
,
partition pcbp697 values less than ('SM
CASE','Brand#521')
,
partition pcbp698 values less than ('SM
CASE','Brand#531')
,
partition pcbp699 values less than ('SM
CASE','Brand#541')
,
partition pcbp700 values less than ('SM
CASE','Brand#551')
,
partition pcbp701 values less than ('SM
DRUM','Brand#111')
,
partition pcbp702 values less than ('SM
DRUM','Brand#121')
,
partition pcbp703 values less than ('SM
DRUM','Brand#131')
,
partition pcbp704 values less than ('SM
DRUM','Brand#141')
,
partition pcbp705 values less than ('SM
DRUM','Brand#151')
,
partition pcbp706 values less than ('SM
DRUM','Brand#211')
,
partition pcbp707 values less than ('SM
DRUM','Brand#221')
,
partition pcbp708 values less than ('SM
DRUM','Brand#231')
,
partition pcbp709 values less than ('SM
DRUM','Brand#241')
,
partition pcbp710 values less than ('SM
DRUM','Brand#251')
,
partition pcbp711 values less than ('SM
DRUM','Brand#311')
,
partition pcbp712 values less than ('SM
DRUM','Brand#321')
,
partition pcbp713 values less than ('SM
DRUM','Brand#331')
,
partition pcbp714 values less than ('SM
DRUM','Brand#341')
,
partition pcbp715 values less than ('SM
DRUM','Brand#351')
,
partition pcbp716 values less than ('SM
DRUM','Brand#411')
,
partition pcbp717 values less than ('SM
DRUM','Brand#421')
,
partition pcbp718 values less than ('SM
DRUM','Brand#431')
```

```
,
partition pcbp719 values less than ('SM
DRUM','Brand#441')
,
partition pcbp720 values less than ('SM
DRUM','Brand#451')
,
partition pcbp721 values less than ('SM
DRUM','Brand#511')
,
partition pcbp722 values less than ('SM
DRUM','Brand#521')
,
partition pcbp723 values less than ('SM
DRUM','Brand#531')
,
partition pcbp724 values less than ('SM
DRUM','Brand#541')
,
partition pcbp725 values less than ('SM
DRUM','Brand#551')
,
partition pcbp726 values less than ('SM
JAR','Brand#111')
,
partition pcbp727 values less than ('SM
JAR','Brand#121')
,
partition pcbp728 values less than ('SM
JAR','Brand#131')
,
partition pcbp729 values less than ('SM
JAR','Brand#141')
,
partition pcbp730 values less than ('SM
JAR','Brand#151')
,
partition pcbp731 values less than ('SM
JAR','Brand#211')
,
partition pcbp732 values less than ('SM
JAR','Brand#221')
,
partition pcbp733 values less than ('SM
JAR','Brand#231')
,
partition pcbp734 values less than ('SM
JAR','Brand#241')
,
partition pcbp735 values less than ('SM
JAR','Brand#251')
,
partition pcbp736 values less than ('SM
JAR','Brand#311')
,
partition pcbp737 values less than ('SM
JAR','Brand#321')
,
partition pcbp738 values less than ('SM
JAR','Brand#331')
,
partition pcbp739 values less than ('SM
JAR','Brand#341')
,
partition pcbp740 values less than ('SM
JAR','Brand#351')
,
partition pcbp741 values less than ('SM
JAR','Brand#411')
,
partition pcbp742 values less than ('SM
JAR','Brand#421')
,
partition pcbp743 values less than ('SM
JAR','Brand#431')
,
partition pcbp744 values less than ('SM
JAR','Brand#441')
,
partition pcbp745 values less than ('SM
JAR','Brand#451')
,
partition pcbp746 values less than ('SM
JAR','Brand#511')
,
partition pcbp747 values less than ('SM
JAR','Brand#521')
,
partition pcbp748 values less than ('SM
JAR','Brand#531')
,
partition pcbp749 values less than ('SM
JAR','Brand#541')
,
partition pcbp750 values less than ('SM
JAR','Brand#551')
,
partition pcbp751 values less than ('SM
PACK','Brand#111')
,
partition pcbp752 values less than ('SM
PACK','Brand#121')
,
partition pcbp753 values less than ('SM
PACK','Brand#131')
,
partition pcbp754 values less than ('SM
PACK','Brand#141')
,
partition pcbp755 values less than ('SM
PACK','Brand#151')
,
partition pcbp756 values less than ('SM
PACK','Brand#211')
,
partition pcbp757 values less than ('SM
PACK','Brand#221')
,
partition pcbp758 values less than ('SM
PACK','Brand#231')
,
partition pcbp759 values less than ('SM
PACK','Brand#241')
,
partition pcbp760 values less than ('SM
PACK','Brand#251')
,
partition pcbp761 values less than ('SM
PACK','Brand#311')
,
partition pcbp762 values less than ('SM
PACK','Brand#321')
,
partition pcbp763 values less than ('SM
PACK','Brand#331')
,
partition pcbp764 values less than ('SM
PACK','Brand#341')
```

```
,
partition pcbp765 values less than ('SM
PACK','Brand#351')
,
partition pcbp766 values less than ('SM
PACK','Brand#411')
,
partition pcbp767 values less than ('SM
PACK','Brand#421')
,
partition pcbp768 values less than ('SM
PACK','Brand#431')
,
partition pcbp769 values less than ('SM
PACK','Brand#441')
,
partition pcbp770 values less than ('SM
PACK','Brand#451')
,
partition pcbp771 values less than ('SM
PACK','Brand#511')
,
partition pcbp772 values less than ('SM
PACK','Brand#521')
,
partition pcbp773 values less than ('SM
PACK','Brand#531')
,
partition pcbp774 values less than ('SM
PACK','Brand#541')
,
partition pcbp775 values less than ('SM
PACK','Brand#551')
,
partition pcbp776 values less than ('SM
PKG','Brand#111')
,
partition pcbp777 values less than ('SM
PKG','Brand#121')
,
partition pcbp778 values less than ('SM
PKG','Brand#131')
,
partition pcbp779 values less than ('SM
PKG','Brand#141')
,
partition pcbp780 values less than ('SM
PKG','Brand#151')
,
partition pcbp781 values less than ('SM
PKG','Brand#211')
,
partition pcbp782 values less than ('SM
PKG','Brand#221')
,
partition pcbp783 values less than ('SM
PKG','Brand#231')
,
partition pcbp784 values less than ('SM
PKG','Brand#241')
,
partition pcbp785 values less than ('SM
PKG','Brand#251')
,
partition pcbp786 values less than ('SM
PKG','Brand#311')
,
partition pcbp787 values less than ('SM
PKG','Brand#321')
,
partition pcbp788 values less than ('SM
PKG','Brand#331')
,
partition pcbp789 values less than ('SM
PKG','Brand#341')
,
partition pcbp790 values less than ('SM
PKG','Brand#351')
,
partition pcbp791 values less than ('SM
PKG','Brand#411')
,
partition pcbp792 values less than ('SM
PKG','Brand#421')
,
partition pcbp793 values less than ('SM
PKG','Brand#431')
,
partition pcbp794 values less than ('SM
PKG','Brand#441')
,
partition pcbp795 values less than ('SM
PKG','Brand#451')
,
partition pcbp796 values less than ('SM
PKG','Brand#511')
,
partition pcbp797 values less than ('SM
PKG','Brand#521')
,
partition pcbp798 values less than ('SM
PKG','Brand#531')
,
partition pcbp799 values less than ('SM
PKG','Brand#541')
,
partition pcbp800 values less than ('SM
PKG','Brand#551')
,
partition pcbp801 values less than ('WRAP
BAG','Brand#111')
,
partition pcbp802 values less than ('WRAP
BAG','Brand#121')
,
partition pcbp803 values less than ('WRAP
BAG','Brand#131')
,
partition pcbp804 values less than ('WRAP
BAG','Brand#141')
,
partition pcbp805 values less than ('WRAP
BAG','Brand#151')
,
partition pcbp806 values less than ('WRAP
BAG','Brand#211')
,
partition pcbp807 values less than ('WRAP
BAG','Brand#221')
,
partition pcbp808 values less than ('WRAP
BAG','Brand#231')
,
partition pcbp809 values less than ('WRAP
BAG','Brand#241')
,
partition pcbp810 values less than ('WRAP
BAG','Brand#251')
```

```
,
partition pcbp811 values less than ('WRAP
BAG','Brand#311')
,
partition pcbp812 values less than ('WRAP
BAG','Brand#321')
,
partition pcbp813 values less than ('WRAP
BAG','Brand#331')
,
partition pcbp814 values less than ('WRAP
BAG','Brand#341')
,
partition pcbp815 values less than ('WRAP
BAG','Brand#351')
,
partition pcbp816 values less than ('WRAP
BAG','Brand#411')
,
partition pcbp817 values less than ('WRAP
BAG','Brand#421')
,
partition pcbp818 values less than ('WRAP
BAG','Brand#431')
,
partition pcbp819 values less than ('WRAP
BAG','Brand#441')
,
partition pcbp820 values less than ('WRAP
BAG','Brand#451')
,
partition pcbp821 values less than ('WRAP
BAG','Brand#511')
,
partition pcbp822 values less than ('WRAP
BAG','Brand#521')
,
partition pcbp823 values less than ('WRAP
BAG','Brand#531')
,
partition pcbp824 values less than ('WRAP
BAG','Brand#541')
,
partition pcbp825 values less than ('WRAP
BAG','Brand#551')
,
partition pcbp826 values less than ('WRAP
BOX','Brand#111')
,
partition pcbp827 values less than ('WRAP
BOX','Brand#121')
,
partition pcbp828 values less than ('WRAP
BOX','Brand#131')
,
partition pcbp829 values less than ('WRAP
BOX','Brand#141')
,
partition pcbp830 values less than ('WRAP
BOX','Brand#151')
,
partition pcbp831 values less than ('WRAP
BOX','Brand#211')
,
partition pcbp832 values less than ('WRAP
BOX','Brand#221')
,
partition pcbp833 values less than ('WRAP
BOX','Brand#231')
,
partition pcbp834 values less than ('WRAP
BOX','Brand#241')
,
partition pcbp835 values less than ('WRAP
BOX','Brand#251')
,
partition pcbp836 values less than ('WRAP
BOX','Brand#311')
,
partition pcbp837 values less than ('WRAP
BOX','Brand#321')
,
partition pcbp838 values less than ('WRAP
BOX','Brand#331')
,
partition pcbp839 values less than ('WRAP
BOX','Brand#341')
,
partition pcbp840 values less than ('WRAP
BOX','Brand#351')
,
partition pcbp841 values less than ('WRAP
BOX','Brand#411')
,
partition pcbp842 values less than ('WRAP
BOX','Brand#421')
,
partition pcbp843 values less than ('WRAP
BOX','Brand#431')
,
partition pcbp844 values less than ('WRAP
BOX','Brand#441')
,
partition pcbp845 values less than ('WRAP
BOX','Brand#451')
,
partition pcbp846 values less than ('WRAP
BOX','Brand#511')
,
partition pcbp847 values less than ('WRAP
BOX','Brand#521')
,
partition pcbp848 values less than ('WRAP
BOX','Brand#531')
,
partition pcbp849 values less than ('WRAP
BOX','Brand#541')
,
partition pcbp850 values less than ('WRAP
BOX','Brand#551')
,
partition pcbp851 values less than ('WRAP
CAN','Brand#111')
,
partition pcbp852 values less than ('WRAP
CAN','Brand#121')
,
partition pcbp853 values less than ('WRAP
CAN','Brand#131')
,
partition pcbp854 values less than ('WRAP
CAN','Brand#141')
,
partition pcbp855 values less than ('WRAP
CAN','Brand#151')
,
partition pcbp856 values less than ('WRAP
CAN','Brand#211')
```

```
,
partition pcbp857 values less than ('WRAP
CAN','Brand#221')
,
partition pcbp858 values less than ('WRAP
CAN','Brand#231')
,
partition pcbp859 values less than ('WRAP
CAN','Brand#241')
,
partition pcbp860 values less than ('WRAP
CAN','Brand#251')
,
partition pcbp861 values less than ('WRAP
CAN','Brand#311')
,
partition pcbp862 values less than ('WRAP
CAN','Brand#321')
,
partition pcbp863 values less than ('WRAP
CAN','Brand#331')
,
partition pcbp864 values less than ('WRAP
CAN','Brand#341')
,
partition pcbp865 values less than ('WRAP
CAN','Brand#351')
,
partition pcbp866 values less than ('WRAP
CAN','Brand#411')
,
partition pcbp867 values less than ('WRAP
CAN','Brand#421')
,
partition pcbp868 values less than ('WRAP
CAN','Brand#431')
,
partition pcbp869 values less than ('WRAP
CAN','Brand#441')
,
partition pcbp870 values less than ('WRAP
CAN','Brand#451')
,
partition pcbp871 values less than ('WRAP
CAN','Brand#511')
,
partition pcbp872 values less than ('WRAP
CAN','Brand#521')
,
partition pcbp873 values less than ('WRAP
CAN','Brand#531')
,
partition pcbp874 values less than ('WRAP
CAN','Brand#541')
,
partition pcbp875 values less than ('WRAP
CAN','Brand#551')
,
partition pcbp876 values less than ('WRAP
CASE','Brand#111')
,
partition pcbp877 values less than ('WRAP
CASE','Brand#121')
,
partition pcbp878 values less than ('WRAP
CASE','Brand#131')
,
partition pcbp879 values less than ('WRAP
CASE','Brand#141')
,
partition pcbp880 values less than ('WRAP
CASE','Brand#151')
,
partition pcbp881 values less than ('WRAP
CASE','Brand#211')
,
partition pcbp882 values less than ('WRAP
CASE','Brand#221')
,
partition pcbp883 values less than ('WRAP
CASE','Brand#231')
,
partition pcbp884 values less than ('WRAP
CASE','Brand#241')
,
partition pcbp885 values less than ('WRAP
CASE','Brand#251')
,
partition pcbp886 values less than ('WRAP
CASE','Brand#311')
,
partition pcbp887 values less than ('WRAP
CASE','Brand#321')
,
partition pcbp888 values less than ('WRAP
CASE','Brand#331')
,
partition pcbp889 values less than ('WRAP
CASE','Brand#341')
,
partition pcbp890 values less than ('WRAP
CASE','Brand#351')
,
partition pcbp891 values less than ('WRAP
CASE','Brand#411')
,
partition pcbp892 values less than ('WRAP
CASE','Brand#421')
,
partition pcbp893 values less than ('WRAP
CASE','Brand#431')
,
partition pcbp894 values less than ('WRAP
CASE','Brand#441')
,
partition pcbp895 values less than ('WRAP
CASE','Brand#451')
,
partition pcbp896 values less than ('WRAP
CASE','Brand#511')
,
partition pcbp897 values less than ('WRAP
CASE','Brand#521')
,
partition pcbp898 values less than ('WRAP
CASE','Brand#531')
,
partition pcbp899 values less than ('WRAP
CASE','Brand#541')
,
partition pcbp900 values less than ('WRAP
CASE','Brand#551')
,
partition pcbp901 values less than ('WRAP
DRUM','Brand#111')
,
partition pcbp902 values less than ('WRAP
DRUM','Brand#121')
```

```
,
partition pcbp903 values less than ('WRAP
DRUM','Brand#131')
,
partition pcbp904 values less than ('WRAP
DRUM','Brand#141')
,
partition pcbp905 values less than ('WRAP
DRUM','Brand#151')
,
partition pcbp906 values less than ('WRAP
DRUM','Brand#211')
,
partition pcbp907 values less than ('WRAP
DRUM','Brand#221')
,
partition pcbp908 values less than ('WRAP
DRUM','Brand#231')
,
partition pcbp909 values less than ('WRAP
DRUM','Brand#241')
,
partition pcbp910 values less than ('WRAP
DRUM','Brand#251')
,
partition pcbp911 values less than ('WRAP
DRUM','Brand#311')
,
partition pcbp912 values less than ('WRAP
DRUM','Brand#321')
,
partition pcbp913 values less than ('WRAP
DRUM','Brand#331')
,
partition pcbp914 values less than ('WRAP
DRUM','Brand#341')
,
partition pcbp915 values less than ('WRAP
DRUM','Brand#351')
,
partition pcbp916 values less than ('WRAP
DRUM','Brand#411')
,
partition pcbp917 values less than ('WRAP
DRUM','Brand#421')
,
partition pcbp918 values less than ('WRAP
DRUM','Brand#431')
,
partition pcbp919 values less than ('WRAP
DRUM','Brand#441')
,
partition pcbp920 values less than ('WRAP
DRUM','Brand#451')
,
partition pcbp921 values less than ('WRAP
DRUM','Brand#511')
,
partition pcbp922 values less than ('WRAP
DRUM','Brand#521')
,
partition pcbp923 values less than ('WRAP
DRUM','Brand#531')
,
partition pcbp924 values less than ('WRAP
DRUM','Brand#541')
,
partition pcbp925 values less than ('WRAP
DRUM','Brand#551')
,
partition pcbp926 values less than ('WRAP
JAR','Brand#111')
,
partition pcbp927 values less than ('WRAP
JAR','Brand#121')
,
partition pcbp928 values less than ('WRAP
JAR','Brand#131')
,
partition pcbp929 values less than ('WRAP
JAR','Brand#141')
,
partition pcbp930 values less than ('WRAP
JAR','Brand#151')
,
partition pcbp931 values less than ('WRAP
JAR','Brand#211')
,
partition pcbp932 values less than ('WRAP
JAR','Brand#221')
,
partition pcbp933 values less than ('WRAP
JAR','Brand#231')
,
partition pcbp934 values less than ('WRAP
JAR','Brand#241')
,
partition pcbp935 values less than ('WRAP
JAR','Brand#251')
,
partition pcbp936 values less than ('WRAP
JAR','Brand#311')
,
partition pcbp937 values less than ('WRAP
JAR','Brand#321')
,
partition pcbp938 values less than ('WRAP
JAR','Brand#331')
,
partition pcbp939 values less than ('WRAP
JAR','Brand#341')
,
partition pcbp940 values less than ('WRAP
JAR','Brand#351')
,
partition pcbp941 values less than ('WRAP
JAR','Brand#411')
,
partition pcbp942 values less than ('WRAP
JAR','Brand#421')
,
partition pcbp943 values less than ('WRAP
JAR','Brand#431')
,
partition pcbp944 values less than ('WRAP
JAR','Brand#441')
,
partition pcbp945 values less than ('WRAP
JAR','Brand#451')
,
partition pcbp946 values less than ('WRAP
JAR','Brand#511')
,
partition pcbp947 values less than ('WRAP
JAR','Brand#521')
,
partition pcbp948 values less than ('WRAP
JAR','Brand#531')
```

```
,
partition pcbp949 values less than ('WRAP
JAR','Brand#541')
,
partition pcbp950 values less than ('WRAP
JAR','Brand#551')
,
partition pcbp951 values less than ('WRAP
PACK','Brand#111')
,
partition pcbp952 values less than ('WRAP
PACK','Brand#121')
,
partition pcbp953 values less than ('WRAP
PACK','Brand#131')
,
partition pcbp954 values less than ('WRAP
PACK','Brand#141')
,
partition pcbp955 values less than ('WRAP
PACK','Brand#151')
,
partition pcbp956 values less than ('WRAP
PACK','Brand#211')
,
partition pcbp957 values less than ('WRAP
PACK','Brand#221')
,
partition pcbp958 values less than ('WRAP
PACK','Brand#231')
,
partition pcbp959 values less than ('WRAP
PACK','Brand#241')
,
partition pcbp960 values less than ('WRAP
PACK','Brand#251')
,
partition pcbp961 values less than ('WRAP
PACK','Brand#311')
,
partition pcbp962 values less than ('WRAP
PACK','Brand#321')
,
partition pcbp963 values less than ('WRAP
PACK','Brand#331')
,
partition pcbp964 values less than ('WRAP
PACK','Brand#341')
,
partition pcbp965 values less than ('WRAP
PACK','Brand#351')
,
partition pcbp966 values less than ('WRAP
PACK','Brand#411')
,
partition pcbp967 values less than ('WRAP
PACK','Brand#421')
,
partition pcbp968 values less than ('WRAP
PACK','Brand#431')
,
partition pcbp969 values less than ('WRAP
PACK','Brand#441')
,
partition pcbp970 values less than ('WRAP
PACK','Brand#451')
,
partition pcbp971 values less than ('WRAP
PACK','Brand#511')
,
partition pcbp972 values less than ('WRAP
PACK','Brand#521')
,
partition pcbp973 values less than ('WRAP
PACK','Brand#531')
,
partition pcbp974 values less than ('WRAP
PACK','Brand#541')
,
partition pcbp975 values less than ('WRAP
PACK','Brand#551')
,
partition pcbp976 values less than ('WRAP
PKG','Brand#111')
,
partition pcbp977 values less than ('WRAP
PKG','Brand#121')
,
partition pcbp978 values less than ('WRAP
PKG','Brand#131')
,
partition pcbp979 values less than ('WRAP
PKG','Brand#141')
,
partition pcbp980 values less than ('WRAP
PKG','Brand#151')
,
partition pcbp981 values less than ('WRAP
PKG','Brand#211')
,
partition pcbp982 values less than ('WRAP
PKG','Brand#221')
,
partition pcbp983 values less than ('WRAP
PKG','Brand#231')
,
partition pcbp984 values less than ('WRAP
PKG','Brand#241')
,
partition pcbp985 values less than ('WRAP
PKG','Brand#251')
,
partition pcbp986 values less than ('WRAP
PKG','Brand#311')
,
partition pcbp987 values less than ('WRAP
PKG','Brand#321')
,
partition pcbp988 values less than ('WRAP
PKG','Brand#331')
,
partition pcbp989 values less than ('WRAP
PKG','Brand#341')
,
partition pcbp990 values less than ('WRAP
PKG','Brand#351')
,
partition pcbp991 values less than ('WRAP
PKG','Brand#411')
,
partition pcbp992 values less than ('WRAP
PKG','Brand#421')
,
partition pcbp993 values less than ('WRAP
PKG','Brand#431')
,
partition pcbp994 values less than ('WRAP
PKG','Brand#441')
```

```
,
partition pcbp995 values less than ('WRAP
PKG','Brand#451')
,
partition pcbp996 values less than ('WRAP
PKG','Brand#511')
,
partition pcbp997 values less than ('WRAP
PKG','Brand#521')
,
partition pcbp998 values less than ('WRAP
PKG','Brand#531')
,
partition pcbp999 values less than ('WRAP
PKG','Brand#541')
,
partition pcbp1000 values less than
(MAXVALUE,MAXVALUE)
)
;
}
*time=Begin creating index i_n_nationkey
*sql
{
connect tpcd/tpcd;
create unique index i_n_nationkey
on nation (n_nationkey)
pctfree 0
nologging
compute statistics
tablespace ts_constr
storage (initial 32k next 32k pctincrease 0)
;
}
*time=Begin creating index i_r_regionkey
*sql
{
connect tpcd/tpcd;
create unique index i_r_regionkey
on region (r_regionkey)
pctfree 0
nologging
compute statistics
tablespace ts_constr
storage (initial 32k next 32k pctincrease 0)
;
}
*time=Begin creating index i_s_suppkey
*sql
{
connect tpcd/tpcd;
create unique index i_s_suppkey
on supplier (s_suppkey)
pctfree 0
nologging
compute statistics
tablespace ts_constr
storage (initial 1m next 1m pctincrease 0)
parallel (degree 16)
;
}
*time=Begin creating index i_c_custkey
*sql
{
connect tpcd/tpcd;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 0
nologging
compute statistics
tablespace ts_constr
storage (initial 13m next 13m pctincrease 0)
parallel (degree 16)
;
}
*time=Begin creating index i_ps_partkey_suppkey
*sql
{
connect tpcd/tpcd;
create unique index i_ps_partkey_suppkey
on partsupp (ps_partkey,ps_suppkey)
pctfree 0
nologging
compute statistics
tablespace ts_constr
storage (initial 70m next 70m pctincrease 0)
parallel (degree 16)
;
}
*time=Begin creating index i_p_partkey
*sql
{
connect tpcd/tpcd;
create unique index i_p_partkey
on parts (p_partkey)
pctfree 0
nologging
compute statistics
tablespace ts_constr
storage (initial 18m next 18m pctincrease 0)
parallel (degree 16)
;
}
*time=Begin creating constraint rpk_r_regionkey
*sql
{
connect tpcd/tpcd;
alter table region add constraint rpk_r_regionkey
      primary key (r_regionkey)
;
}
*wait
*time=Done creating constraint rpk_r_regionkey
*time=Begin creating constraint npk_n_nationkey
*sql
{
connect tpcd/tpcd;
alter table nation add constraint npk_n_nationkey
      primary key (n_nationkey)
;
}
*wait
*time=Done creating constraint npk_n_nationkey
*time=Begin creating constraint spk_s_suppkey
*sql
{
connect tpcd/tpcd;
alter table supplier add constraint spk_s_suppkey
      primary key (s_suppkey)
;
}
*wait
*time=Done creating constraint spk_s_suppkey
*time=Begin creating constraint cpk_c_custkey
*sql
{
connect tpcd/tpcd;
alter table customer add constraint cpk_c_custkey
```

```
      primary key (c_custkey)
;
}
*wait
*time=Done creating constraint cpk_c_custkey
*time=Begin creating constraint ppk_p_partkey
*sql
{
connect tpcd/tpcd;
alter table parts add constraint ppk_p_partkey
      primary key (p_partkey)
;
}
*wait
*time=Done creating constraint ppk_p_partkey
*time=Begin creating constraint
pspk_ps_partkey_suppkey
*sql
{
connect tpcd/tpcd;
alter table partsupp add constraint
pspk_ps_partkey_suppkey
      primary key (ps_partkey,ps_suppkey)
;
}
*wait
*time=Done creating constraint
pspk_ps_partkey_suppkey
*time=Begin creating constraint ook_o_orderkey
*sql
{
connect tpcd/tpcd;
alter table orders add constraint ook_o_orderkey
      primary key (o_orderkey)
;
}
*wait
*time=Done creating constraint ook_o_orderkey
*wait
*time=Alter DOP of indexes
*sql
{
connect tpcd/tpcd;
}
*wait
*time=Done altering DOP of indexes
*time=Done index creation
*wait
*bgoff
%e-ixcre
```

100g_816_plcre.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-plcre
*bgon=1
################################################
############################
# NT Raw Partition and Setlink Creation Phase
*time=Begin NT Raw Partition and Setlink creation
*sh
{
creapart -d PhysicalDrive3
}
*time=Deleted partitions on PhysicalDrive3
*sh
{
creapart -x PhysicalDrive3
}
*time=Created extended partition on
PhysicalDrive3
*sh
{
creapart -d PhysicalDrive4
}
*time=Deleted partitions on PhysicalDrive4
*sh
{
creapart -x PhysicalDrive4
}
*time=Created extended partition on
PhysicalDrive4
*sh
{
creapart -d PhysicalDrive9
}
*time=Deleted partitions on PhysicalDrive9
*sh
{
creapart -x PhysicalDrive9
}
*time=Created extended partition on
PhysicalDrive9
*sh
{
creapart -d PhysicalDrive10
}
*time=Deleted partitions on PhysicalDrive10
*sh
{
creapart -x PhysicalDrive10
}
*time=Created extended partition on
PhysicalDrive10
*sh
```

```
{
creapart -d PhysicalDrive11
}
*time=Deleted partitions on PhysicalDrive11
*sh
{
creapart -x PhysicalDrive11
}
*time=Created extended partition on
PhysicalDrive11
*sh
{
creapart -d PhysicalDrive6
}
*time=Deleted partitions on PhysicalDrive6
*sh
{
creapart -x PhysicalDrive6
}
*time=Created extended partition on
PhysicalDrive6
*sh
{
creapart -l PhysicalDrive3 30
}
*time='e:controlcntr100g1'
        \device\harddisk3\partition1
*sh
{
creapart -l PhysicalDrive3 30
}
*time='d:controlcntr100g2'
        \device\harddisk3\partition2
*sh
{
creapart -l PhysicalDrive6 3201
}
*time=log_100g1   \device\harddisk6\partition1
*sh
{
creapart -l PhysicalDrive6 3201
}
*time=log_100g2   \device\harddisk6\partition2
*sh
{
creapart -l PhysicalDrive3 21
}
*time=sys_100g1   \device\harddisk3\partition3
*sh
{
creapart -l PhysicalDrive4 21
}
*time=sys_100g2   \device\harddisk4\partition1
*sh
{
creapart -l PhysicalDrive9 21
}
*time=sys_100g3   \device\harddisk9\partition1
*sh
{
creapart -l PhysicalDrive10 21
}
*time=sys_100g4   \device\harddisk10\partition1
*sh
{
creapart -l PhysicalDrive11 21
}
*time=sys_100g5   \device\harddisk11\partition1
*sh
{
creapart -l PhysicalDrive3 1001
}
*time=undo_100g1  \device\harddisk3\partition4
*sh
{
creapart -l PhysicalDrive4 1001
}
*time=undo_100g2  \device\harddisk4\partition2
*sh
{
creapart -l PhysicalDrive9 1001
}
*time=undo_100g3  \device\harddisk9\partition2
*sh
{
creapart -l PhysicalDrive10 1001
}
*time=undo_100g4  \device\harddisk10\partition2
*sh
{
creapart -l PhysicalDrive11 1001
}
*time=undo_100g5  \device\harddisk11\partition2
*sh
{
creapart -l PhysicalDrive3 20481
}
*time=ts_temp_100g1
        \device\harddisk3\partition5
*sh
{
creapart -l PhysicalDrive4 20481
}
*time=ts_temp_100g2
        \device\harddisk4\partition3
*sh
{
creapart -l PhysicalDrive9 20481
}
*time=ts_temp_100g3
        \device\harddisk9\partition3
*sh
{
creapart -l PhysicalDrive10 20481
}
*time=ts_temp_100g4
        \device\harddisk10\partition3
*sh
{
creapart -l PhysicalDrive11 20481
}
*time=ts_temp_100g5
        \device\harddisk11\partition3
*sh
{
creapart -l PhysicalDrive3 20481
}
*time=ts_temp_100g6
        \device\harddisk3\partition6
*sh
{
creapart -l PhysicalDrive4 20481
}
*time=ts_temp_100g7
        \device\harddisk4\partition4
*sh
{
creapart -l PhysicalDrive9 20481
```

```
}
*time=ts_temp_100g8
        \device\harddisk9\partition4
*sh
{
creapart -l PhysicalDrive10 20481
}
*time=ts_temp_100g9
        \device\harddisk10\partition4
*sh
{
creapart -l PhysicalDrive11 20481
}
*time=ts_temp_100g10
        \device\harddisk11\partition4
*sh
{
creapart -l PhysicalDrive3 2001
}
*time=ts_default_100g1
        \device\harddisk3\partition7
*sh
{
creapart -l PhysicalDrive4 2001
}
*time=ts_default_100g2
        \device\harddisk4\partition5
*sh
{
creapart -l PhysicalDrive9 2001
}
*time=ts_default_100g3
        \device\harddisk9\partition5
*sh
{
creapart -l PhysicalDrive10 2001
}
*time=ts_default_100g4
        \device\harddisk10\partition5
*sh
{
creapart -l PhysicalDrive11 2001
}
*time=ts_default_100g5
        \device\harddisk11\partition5
*sh
{
creapart -l PhysicalDrive3 16092
}
*time=ts_l_100g1  \device\harddisk3\partition8
*sh
{
creapart -l PhysicalDrive4 16092
}
*time=ts_l_100g2  \device\harddisk4\partition6
*sh
{
creapart -l PhysicalDrive9 16092
}
*time=ts_l_100g3  \device\harddisk9\partition6
*sh
{
creapart -l PhysicalDrive10 16092
}
*time=ts_l_100g4  \device\harddisk10\partition6
*sh
{
creapart -l PhysicalDrive11 16092
}
*time=ts_l_100g5  \device\harddisk11\partition6
*sh
{
creapart -l PhysicalDrive3 512
}
*time=ts_c_100g1  \device\harddisk3\partition9
*sh
{
creapart -l PhysicalDrive4 512
}
*time=ts_c_100g2  \device\harddisk4\partition7
*sh
{
creapart -l PhysicalDrive9 512
}
*time=ts_c_100g3  \device\harddisk9\partition7
*sh
{
creapart -l PhysicalDrive10 512
}
*time=ts_c_100g4  \device\harddisk10\partition7
*sh
{
creapart -l PhysicalDrive11 512
}
*time=ts_c_100g5  \device\harddisk11\partition7
*sh
{
creapart -l PhysicalDrive3 6001
}
*time=ts_o_100g1  \device\harddisk3\partition10
*sh
{
creapart -l PhysicalDrive4 6001
}
*time=ts_o_100g2  \device\harddisk4\partition8
*sh
{
creapart -l PhysicalDrive9 6001
}
*time=ts_o_100g3  \device\harddisk9\partition8
*sh
{
creapart -l PhysicalDrive10 6001
}
*time=ts_o_100g4  \device\harddisk10\partition8
*sh
{
creapart -l PhysicalDrive11 6001
}
*time=ts_o_100g5  \device\harddisk11\partition8
*sh
{
creapart -l PhysicalDrive3 606
}
*time=ts_p_100g1  \device\harddisk3\partition11
*sh
{
creapart -l PhysicalDrive4 606
}
*time=ts_p_100g2  \device\harddisk4\partition9
*sh
{
creapart -l PhysicalDrive9 606
}
*time=ts_p_100g3  \device\harddisk9\partition9
*sh
{
creapart -l PhysicalDrive10 606
```

```
}
*time=ts_p_100g4  \device\harddisk10\partition9
*sh
{
creapart -l PhysicalDrive11 606
}
*time=ts_p_100g5  \device\harddisk11\partition9
*sh
{
creapart -l PhysicalDrive3 2369
}
*time=ts_ps_100g1 \device\harddisk3\partition12
*sh
{
creapart -l PhysicalDrive4 2369
}
*time=ts_ps_100g2 \device\harddisk4\partition10
*sh
{
creapart -l PhysicalDrive9 2369
}
*time=ts_ps_100g3 \device\harddisk9\partition10
*sh
{
creapart -l PhysicalDrive10 2369
}
*time=ts_ps_100g4 \device\harddisk10\partition10
*sh
{
creapart -l PhysicalDrive11 2369
}
*time=ts_ps_100g5 \device\harddisk11\partition10
*sh
{
creapart -l PhysicalDrive3 32
}
*time=ts_s_100g1  \device\harddisk3\partition13
*sh
{
creapart -l PhysicalDrive4 32
}
*time=ts_s_100g2  \device\harddisk4\partition11
*sh
{
creapart -l PhysicalDrive9 32
}
*time=ts_s_100g3  \device\harddisk9\partition11
*sh
{
creapart -l PhysicalDrive10 32
}
*time=ts_s_100g4  \device\harddisk10\partition11
*sh
{
creapart -l PhysicalDrive11 32
}
*time=ts_s_100g5  \device\harddisk11\partition11
*sh
{
creapart -l PhysicalDrive3 4
}
*time=ts_ma_lineitem_100g1
        \device\harddisk3\partition14
*sh
{
creapart -l PhysicalDrive4 4
}
*time=ts_ma_lineitem_100g2
        \device\harddisk4\partition12
*sh
{
creapart -l PhysicalDrive9 4
}
*time=ts_ma_lineitem_100g3
        \device\harddisk9\partition12
*sh
{
creapart -l PhysicalDrive10 4
}
*time=ts_ma_lineitem_100g4
        \device\harddisk10\partition12
*sh
{
creapart -l PhysicalDrive11 4
}
*time=ts_ma_lineitem_100g5
        \device\harddisk11\partition12
*sh
{
creapart -l PhysicalDrive3 16877
}
*time=ts_mj_locs_100g1
        \device\harddisk3\partition15
*sh
{
creapart -l PhysicalDrive4 16877
}
*time=ts_mj_locs_100g2
        \device\harddisk4\partition13
*sh
{
creapart -l PhysicalDrive9 16877
}
*time=ts_mj_locs_100g3
        \device\harddisk9\partition13
*sh
{
creapart -l PhysicalDrive10 16877
}
*time=ts_mj_locs_100g4
        \device\harddisk10\partition13
*sh
{
creapart -l PhysicalDrive11 16877
}
*time=ts_mj_locs_100g5
        \device\harddisk11\partition13
*sh
{
creapart -l PhysicalDrive3 16877
}
*time=ts_mj_locs_100g6
        \device\harddisk3\partition16
*sh
{
creapart -l PhysicalDrive4 16877
}
*time=ts_mj_locs_100g7
        \device\harddisk4\partition14
*sh
{
creapart -l PhysicalDrive9 16877
}
*time=ts_mj_locs_100g8
        \device\harddisk9\partition14
*sh
{
creapart -l PhysicalDrive10 16877
```

```
}
*time=ts_mj_locs_100g9
        \device\harddisk10\partition14
*sh
{
creapart -l PhysicalDrive11 16877
}
*time=ts_mj_locs_100g10
        \device\harddisk11\partition14
*sh
{
creapart -l PhysicalDrive3 601
}
*time=ts_mllog_100g1
        \device\harddisk3\partition17
*sh
{
creapart -l PhysicalDrive4 601
}
*time=ts_mllog_100g2
        \device\harddisk4\partition15
*sh
{
creapart -l PhysicalDrive9 601
}
*time=ts_mllog_100g3
        \device\harddisk9\partition15
*sh
{
creapart -l PhysicalDrive10 601
}
*time=ts_mllog_100g4
        \device\harddisk10\partition15
*sh
{
creapart -l PhysicalDrive11 601
}
*time=ts_mllog_100g5
        \device\harddisk11\partition15
*sh
{
creapart -l PhysicalDrive3 601
}
*time=ts_molog_100g1
        \device\harddisk3\partition18
*sh
{
creapart -l PhysicalDrive4 601
}
*time=ts_molog_100g2
        \device\harddisk4\partition16
*sh
{
creapart -l PhysicalDrive9 601
}
*time=ts_molog_100g3
        \device\harddisk9\partition16
*sh
{
creapart -l PhysicalDrive10 601
}
*time=ts_molog_100g4
        \device\harddisk10\partition16
*sh
{
creapart -l PhysicalDrive11 601
}
*time=ts_molog_100g5
        \device\harddisk11\partition16
*sh
{
creapart -l PhysicalDrive3 4
}
*time=ts_maint_100g1
        \device\harddisk3\partition19
*sh
{
creapart -l PhysicalDrive4 4
}
*time=ts_maint_100g2
        \device\harddisk4\partition17
*sh
{
creapart -l PhysicalDrive9 4
}
*time=ts_maint_100g3
        \device\harddisk9\partition17
*sh
{
creapart -l PhysicalDrive10 4
}
*time=ts_maint_100g4
        \device\harddisk10\partition17
*sh
{
creapart -l PhysicalDrive11 4
}
*time=ts_maint_100g5
        \device\harddisk11\partition17
*sh
{
creapart -l PhysicalDrive3 3864
}
*time=ts_mj_ppssrn_100g1
        \device\harddisk3\partition20
*sh
{
creapart -l PhysicalDrive4 3864
}
*time=ts_mj_ppssrn_100g2
        \device\harddisk4\partition18
*sh
{
creapart -l PhysicalDrive9 3864
}
*time=ts_mj_ppssrn_100g3
        \device\harddisk9\partition18
*sh
{
creapart -l PhysicalDrive10 3864
}
*time=ts_mj_ppssrn_100g4
        \device\harddisk10\partition18
*sh
{
creapart -l PhysicalDrive11 3864
}
*time=ts_mj_ppssrn_100g5
        \device\harddisk11\partition18
*sh
{
creapart -l PhysicalDrive3 201
}
*time=ts_11       \device\harddisk3\partition21
*sh
{
creapart -l PhysicalDrive4 201
}
```

```
*time=ts_12       \device\harddisk4\partition19
*sh
{
creapart -l PhysicalDrive9 201
}
*time=ts_13       \device\harddisk9\partition19
*sh
{
creapart -l PhysicalDrive10 201
}
*time=ts_14       \device\harddisk10\partition19
*sh
{
creapart -l PhysicalDrive11 201
}
*time=ts_15       \device\harddisk11\partition19
*sh
{
creapart -l PhysicalDrive3 15601
}
*time=ts_ij_l_sdate_100g1
         \device\harddisk3\partition22
*sh
{
creapart -l PhysicalDrive4 15601
}
*time=ts_ij_l_sdate_100g2
         \device\harddisk4\partition20
*sh
{
creapart -l PhysicalDrive9 15601
}
*time=ts_ij_l_sdate_100g3
         \device\harddisk9\partition20
*sh
{
creapart -l PhysicalDrive10 15601
}
*time=ts_ij_l_sdate_100g4
         \device\harddisk10\partition20
*sh
{
creapart -l PhysicalDrive11 15601
}
*time=ts_ij_l_sdate_100g5
         \device\harddisk11\partition20
*sh
{
creapart -l PhysicalDrive3 8701
}
*time=ts_ij_l_rdate_100g1
         \device\harddisk3\partition23
*sh
{
creapart -l PhysicalDrive4 8701
}
*time=ts_ij_l_rdate_100g2
         \device\harddisk4\partition21
*sh
{
creapart -l PhysicalDrive9 8701
}
*time=ts_ij_l_rdate_100g3
         \device\harddisk9\partition21
*sh
{
creapart -l PhysicalDrive10 8701
}
*time=ts_ij_l_rdate_100g4
         \device\harddisk10\partition21
*sh
{
creapart -l PhysicalDrive11 8701
}
*time=ts_ij_l_rdate_100g5
         \device\harddisk11\partition21
*sh
{
creapart -l PhysicalDrive3 14401
}
*time=ts_ij_l_odate_100g1
         \device\harddisk3\partition24
*sh
{
creapart -l PhysicalDrive4 14401
}
*time=ts_ij_l_odate_100g2
         \device\harddisk4\partition22
*sh
{
creapart -l PhysicalDrive9 14401
}
*time=ts_ij_l_odate_100g3
         \device\harddisk9\partition22
*sh
{
creapart -l PhysicalDrive10 14401
}
*time=ts_ij_l_odate_100g4
         \device\harddisk10\partition22
*sh
{
creapart -l PhysicalDrive11 14401
}
*time=ts_ij_l_odate_100g5
         \device\harddisk11\partition22
*sh
{
creapart -l PhysicalDrive3 9601
}
*time=ts_ij_l_pkey_100g1
         \device\harddisk3\partition25
*sh
{
creapart -l PhysicalDrive4 9601
}
*time=ts_ij_l_pkey_100g2
         \device\harddisk4\partition23
*sh
{
creapart -l PhysicalDrive9 9601
}
*time=ts_ij_l_pkey_100g3
         \device\harddisk9\partition23
*sh
{
creapart -l PhysicalDrive10 9601
}
*time=ts_ij_l_pkey_100g4
         \device\harddisk10\partition23
*sh
{
creapart -l PhysicalDrive11 9601
}
*time=ts_ij_l_pkey_100g5
         \device\harddisk11\partition23
*sh
{
```

```
creapart -l PhysicalDrive3 4621
}
*time=ts_ind_lored_100g1
        \device\harddisk3\partition26
*sh
{
creapart -l PhysicalDrive4 4621
}
*time=ts_ind_lored_100g2
        \device\harddisk4\partition24
*sh
{
creapart -l PhysicalDrive9 4621
}
*time=ts_ind_lored_100g3
        \device\harddisk9\partition24
*sh
{
creapart -l PhysicalDrive10 4621
}
*time=ts_ind_lored_100g4
        \device\harddisk10\partition24
*sh
{
creapart -l PhysicalDrive11 4621
}
*time=ts_ind_lored_100g5
        \device\harddisk11\partition24
*sh
{
creapart -l PhysicalDrive3 1849
}
*time=ts_ind_oclokod_100g1
        \device\harddisk3\partition27
*sh
{
creapart -l PhysicalDrive4 1849
}
*time=ts_ind_oclokod_100g2
        \device\harddisk4\partition25
*sh
{
creapart -l PhysicalDrive9 1849
}
*time=ts_ind_oclokod_100g3
        \device\harddisk9\partition25
*sh
{
creapart -l PhysicalDrive10 1849
}
*time=ts_ind_oclokod_100g4
        \device\harddisk10\partition25
*sh
{
creapart -l PhysicalDrive11 1849
}
*time=ts_ind_oclokod_100g5
        \device\harddisk11\partition25
*sh
{
creapart -l PhysicalDrive3 4201
}
*time=ts_ij_lrid_100g1
        \device\harddisk3\partition28
*sh
{
creapart -l PhysicalDrive4 4201
}
*time=ts_ij_lrid_100g2
        \device\harddisk4\partition26
*sh
{
creapart -l PhysicalDrive9 4201
}
*time=ts_ij_lrid_100g3
        \device\harddisk9\partition26
*sh
{
creapart -l PhysicalDrive10 4201
}
*time=ts_ij_lrid_100g4
        \device\harddisk10\partition26
*sh
{
creapart -l PhysicalDrive11 4201
}
*time=ts_ij_lrid_100g5
        \device\harddisk11\partition26
*sh
{
creapart -l PhysicalDrive3 4201
}
*time=ts_ij_orid_100g1
        \device\harddisk3\partition29
*sh
{
creapart -l PhysicalDrive4 4201
}
*time=ts_ij_orid_100g2
        \device\harddisk4\partition27
*sh
{
creapart -l PhysicalDrive9 4201
}
*time=ts_ij_orid_100g3
        \device\harddisk9\partition27
*sh
{
creapart -l PhysicalDrive10 4201
}
*time=ts_ij_orid_100g4
        \device\harddisk10\partition27
*sh
{
creapart -l PhysicalDrive11 4201
}
*time=ts_ij_orid_100g5
        \device\harddisk11\partition27
*sh
{
creapart -l PhysicalDrive3 40
}
*time=ts_ij_nullrid_100g1
        \device\harddisk3\partition30
*sh
{
creapart -l PhysicalDrive4 40
}
*time=ts_ij_nullrid_100g2
        \device\harddisk4\partition28
*sh
{
creapart -l PhysicalDrive9 40
}
*time=ts_ij_nullrid_100g3
        \device\harddisk9\partition28
*sh
{
```

```
creapart -l PhysicalDrive10 40
}
*time=ts_ij_nullrid_100g4
        \device\harddisk10\partition28
*sh
{
creapart -l PhysicalDrive11 40
}
*time=ts_ij_nullrid_100g5
        \device\harddisk11\partition28
*sh
{
creapart -l PhysicalDrive3 1001
}
*time=ts_ind_okey_100g1
        \device\harddisk3\partition31
*sh
{
creapart -l PhysicalDrive4 1001
}
*time=ts_ind_okey_100g2
        \device\harddisk4\partition29
*sh
{
creapart -l PhysicalDrive9 1001
}
*time=ts_ind_okey_100g3
        \device\harddisk9\partition29
*sh
{
creapart -l PhysicalDrive10 1001
}
*time=ts_ind_okey_100g4
        \device\harddisk10\partition29
*sh
{
creapart -l PhysicalDrive11 1001
}
*time=ts_ind_okey_100g5
        \device\harddisk11\partition29
*sh
{
creapart -l PhysicalDrive3 1001
}
*time=ts_constr_100g1
        \device\harddisk3\partition32
*sh
{
creapart -l PhysicalDrive4 1001
}
*time=ts_constr_100g2
        \device\harddisk4\partition30
*sh
{
creapart -l PhysicalDrive9 1001
}
*time=ts_constr_100g3
        \device\harddisk9\partition30
*sh
{
creapart -l PhysicalDrive10 1001
}
*time=ts_constr_100g4
        \device\harddisk10\partition30
*sh
{
creapart -l PhysicalDrive11 1001
}
*time=ts_constr_100g5
        \device\harddisk11\partition30
*sh
{
creapart -l PhysicalDrive3 1561
}
*time=ts_ij_p_ppsn_100g1
        \device\harddisk3\partition33
*sh
{
creapart -l PhysicalDrive4 1561
}
*time=ts_ij_p_ppsn_100g2
        \device\harddisk4\partition31
*sh
{
creapart -l PhysicalDrive9 1561
}
*time=ts_ij_p_ppsn_100g3
        \device\harddisk9\partition31
*sh
{
creapart -l PhysicalDrive10 1561
}
*time=ts_ij_p_ppsn_100g4
        \device\harddisk10\partition31
*sh
{
creapart -l PhysicalDrive11 1561
}
*time=ts_ij_p_ppsn_100g5
        \device\harddisk11\partition31
*sh
{
creapart -l PhysicalDrive3 844
}
*time=ts_ij_p_nnpsps_100g1
        \device\harddisk3\partition34
*sh
{
creapart -l PhysicalDrive4 844
}
*time=ts_ij_p_nnpsps_100g2
        \device\harddisk4\partition32
*sh
{
creapart -l PhysicalDrive9 844
}
*time=ts_ij_p_nnpsps_100g3
        \device\harddisk9\partition32
*sh
{
creapart -l PhysicalDrive10 844
}
*time=ts_ij_p_nnpsps_100g4
        \device\harddisk10\partition32
*sh
{
creapart -l PhysicalDrive11 844
}
*time=ts_ij_p_nnpsps_100g5
        \device\harddisk11\partition32
*sh
{
creapart -l PhysicalDrive3 459
}
*time=ts_p_tp_100g1
        \device\harddisk3\partition35
*sh
{
```

```
creapart -l PhysicalDrive4 459
}
*time=ts_p_tp_100g2
        \device\harddisk4\partition33
*sh
{
creapart -l PhysicalDrive9 459
}
*time=ts_p_tp_100g3
        \device\harddisk9\partition33
*sh
{
creapart -l PhysicalDrive10 459
}
*time=ts_p_tp_100g4
        \device\harddisk10\partition33
*sh
{
creapart -l PhysicalDrive11 459
}
*time=ts_p_tp_100g5
        \device\harddisk11\partition33
*sh
{
creapart -l PhysicalDrive3 3048
}
*time=ts_p_cbp_100g1
        \device\harddisk3\partition36
*sh
{
creapart -l PhysicalDrive4 3048
}
*time=ts_p_cbp_100g2
        \device\harddisk4\partition34
*sh
{
creapart -l PhysicalDrive9 3048
}
*time=ts_p_cbp_100g3
        \device\harddisk9\partition34
*sh
{
creapart -l PhysicalDrive10 3048
}
*time=ts_p_cbp_100g4
        \device\harddisk10\partition34
*sh
{
creapart -l PhysicalDrive11 3048
}
*time=ts_p_cbp_100g5
        \device\harddisk11\partition34
*wait
*time=Done NT Raw Partition creation
*sh
{
setlinks
/F:E:/tpcd814_work/bumpx814/datbat/links100g.prn
}
*wait
*time=Done Setlink links creation
*time=Done NT Raw Partition and Setlink creation
*wait
*bgoff
%e-plcre
```

100g_816_sccre.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-sccre
*bgon=60
################################################
############################
# Schema Creation Phase
*time=Begin schema creation
*time=Cycling database before phase 'sccre'...
*time=Shutting down database...
*sql
{
shutdown normal;
}
*wait
*time=Starting database...
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
}
*wait
*sql
{
drop tablespace ts_default;
create tablespace ts_default
datafile '\\.\ts_default_100g1' size 2000m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_l tablespace
*sql
{
drop tablespace ts_l;
create tablespace ts_l
datafile '\\.\ts_l_100g1' size 16091m reuse
default storage (initial 32k next 31m maxextents
```

```
unlimited pctincrease 0)
;
}
# creating tpcd's ts_c tablespace
*sql
{
drop tablespace ts_c;
create tablespace ts_c
datafile '\\.\ts_c_100g1' size 511m reuse
default storage (initial 32k next 34m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_o tablespace
*sql
{
drop tablespace ts_o;
create tablespace ts_o
datafile '\\.\ts_o_100g1' size 6000m reuse
default storage (initial 32k next 10m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_p tablespace
*sql
{
drop tablespace ts_p;
create tablespace ts_p
datafile '\\.\ts_p_100g1' size 605m reuse
default storage (initial 32k next 4m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ps tablespace
*sql
{
drop tablespace ts_ps;
create tablespace ts_ps
datafile '\\.\ts_ps_100g1' size 2368m reuse
default storage (initial 32k next 789m
pctincrease 0)
;
}
# creating tpcd's ts_s tablespace
*sql
{
drop tablespace ts_s;
create tablespace ts_s
datafile '\\.\ts_s_100g1' size 31m reuse
default storage (initial 32k next 10m pctincrease
0)
;
}
# creating tpcd's ts_ma_lineitem tablespace
*sql
{
drop tablespace ts_ma_lineitem;
create tablespace ts_ma_lineitem
datafile '\\.\ts_ma_lineitem_100g1' size 3m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_mj_locs tablespace
*sql
{
drop tablespace ts_mj_locs;
create tablespace ts_mj_locs
datafile '\\.\ts_mj_locs_100g1' size 16876m reuse
default storage (initial 7m next 7m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_mllog tablespace
*sql
{
drop tablespace ts_mllog;
create tablespace ts_mllog
datafile '\\.\ts_mllog_100g1' size 600m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_molog tablespace
*sql
{
drop tablespace ts_molog;
create tablespace ts_molog
datafile '\\.\ts_molog_100g1' size 600m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_maint tablespace
*sql
{
drop tablespace ts_maint;
create tablespace ts_maint
datafile '\\.\ts_maint_100g1' size 3m reuse
default storage (initial 160k next 160k
maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_mj_ppssrn tablespace
*sql
{
drop tablespace ts_mj_ppssrn;
create tablespace ts_mj_ppssrn
datafile '\\.\ts_mj_ppssrn_100g1' size 3863m
reuse
default storage (initial 4m next 4m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_1 tablespace
*sql
{
drop tablespace ts_1;
create tablespace ts_1
datafile '\\.\ts_11' size 200m reuse
default storage (initial 32k next 64k maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_sdate tablespace
*sql
{
drop tablespace ts_ij_l_sdate;
create tablespace ts_ij_l_sdate
datafile '\\.\ts_ij_l_sdate_100g1' size 15600m
reuse
default storage (initial 4m next 4m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_rdate tablespace
*sql
{
```

```
drop tablespace ts_ij_l_rdate;
create tablespace ts_ij_l_rdate
datafile '\\.\ts_ij_l_rdate_100g1' size 8700m
reuse
default storage (initial 11m next 11m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_odate tablespace
*sql
{
drop tablespace ts_ij_l_odate;
create tablespace ts_ij_l_odate
datafile '\\.\ts_ij_l_odate_100g1' size 14400m
reuse
default storage (initial 18m next 18m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_pkey tablespace
*sql
{
drop tablespace ts_ij_l_pkey;
create tablespace ts_ij_l_pkey
datafile '\\.\ts_ij_l_pkey_100g1' size 9600m
reuse
default storage (initial 12m next 12m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ind_lored tablespace
*sql
{
drop tablespace ts_ind_lored;
create tablespace ts_ind_lored
datafile '\\.\ts_ind_lored_100g1' size 4620m
reuse
default storage (initial 6m next 6m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ind_oclokod tablespace
*sql
{
drop tablespace ts_ind_oclokod;
create tablespace ts_ind_oclokod
datafile '\\.\ts_ind_oclokod_100g1' size 1848m
reuse
default storage (initial 3m next 3m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_lrid tablespace
*sql
{
drop tablespace ts_ij_l_lrid;
create tablespace ts_ij_l_lrid
datafile '\\.\ts_ij_lrid_100g1' size 4200m reuse
default storage (initial 788m next 788m
maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_orid tablespace
*sql
{
drop tablespace ts_ij_l_orid;
create tablespace ts_ij_l_orid
datafile '\\.\ts_ij_orid_100g1' size 4200m reuse
default storage (initial 788m next 788m
maxextents unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_l_nullrid tablespace
*sql
{
drop tablespace ts_ij_l_nullrid;
create tablespace ts_ij_l_nullrid
datafile '\\.\ts_ij_nullrid_100g1' size 39m reuse
default storage (initial 8m next 8m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ind_ookey tablespace
*sql
{
drop tablespace ts_ind_ookey;
create tablespace ts_ind_ookey
datafile '\\.\ts_ind_okey_100g1' size 1000m reuse
default storage (initial 20m next 20m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_constr tablespace
*sql
{
drop tablespace ts_constr;
create tablespace ts_constr
datafile '\\.\ts_constr_100g1' size 1000m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_p_ppsn tablespace
*sql
{
drop tablespace ts_ij_p_ppsn;
create tablespace ts_ij_p_ppsn
datafile '\\.\ts_ij_p_ppsn_100g1' size 1560m
reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_ij_p_nnpsps tablespace
*sql
{
drop tablespace ts_ij_p_nnpsps;
create tablespace ts_ij_p_nnpsps
datafile '\\.\ts_ij_p_nnpsps_100g1' size 843m
reuse
default storage (initial 7m next 7m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_p_tp tablespace
*sql
{
drop tablespace ts_p_tp;
create tablespace ts_p_tp
datafile '\\.\ts_p_tp_100g1' size 458m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_p_cbp tablespace
*sql
{
drop tablespace ts_p_cbp;
```

```
create tablespace ts_p_cbp
datafile '\\.\ts_p_cbp_100g1' size 3047m reuse
default storage (initial 1m next 1m maxextents
unlimited pctincrease 0)
;
}
# creating tpcd's ts_temp tablespace
*sql
{
drop tablespace ts_temp;
create tablespace ts_temp
datafile '\\.\ts_temp_100g1' size 20480m reuse
default storage (initial 11m next 11m maxextents
unlimited pctincrease 0)
;
}
*wait
# adding tpcd's ts_temp datafiles
*sql
{
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g3' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g4' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g5' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g6' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g7' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g8' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g9' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g10' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g11' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g12' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g13' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g14' size 20480m
reuse;
alter tablespace ts_temp
  add datafile '\\.\ts_temp_100g15' size 20480m
reuse;
}
# adding tpcd's ts_mj_locs datafiles
*sql
{
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g3' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g4' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g5' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g6' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g7' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g8' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g9' size 16876m
reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g10' size
16876m reuse;
alter tablespace ts_mj_locs
  add datafile '\\.\ts_mj_locs_100g2' size 16876m
reuse;
}
# adding tpcd's ts_l datafiles
*sql
{
alter tablespace ts_l
  add datafile '\\.\ts_l_100g4' size 16091m
reuse;
alter tablespace ts_l
  add datafile '\\.\ts_l_100g5' size 16091m
reuse;
alter tablespace ts_l
  add datafile '\\.\ts_l_100g2' size 16091m
reuse;
alter tablespace ts_l
  add datafile '\\.\ts_l_100g3' size 16091m
reuse;
}
# adding tpcd's ts_ij_l_sdate datafiles
*sql
{
alter tablespace ts_ij_l_sdate
  add datafile '\\.\ts_ij_l_sdate_100g5' size
15600m reuse;
alter tablespace ts_ij_l_sdate
  add datafile '\\.\ts_ij_l_sdate_100g2' size
15600m reuse;
alter tablespace ts_ij_l_sdate
  add datafile '\\.\ts_ij_l_sdate_100g3' size
15600m reuse;
alter tablespace ts_ij_l_sdate
  add datafile '\\.\ts_ij_l_sdate_100g4' size
15600m reuse;
}
# adding tpcd's ts_ij_l_odate datafiles
*sql
{
alter tablespace ts_ij_l_odate
  add datafile '\\.\ts_ij_l_odate_100g2' size
14400m reuse;
alter tablespace ts_ij_l_odate
  add datafile '\\.\ts_ij_l_odate_100g3' size
14400m reuse;
alter tablespace ts_ij_l_odate
  add datafile '\\.\ts_ij_l_odate_100g4' size
14400m reuse;
alter tablespace ts_ij_l_odate
  add datafile '\\.\ts_ij_l_odate_100g5' size
14400m reuse;
```

```
}
# adding tpcd's ts_ij_l_pkey datafiles
*sql
{
alter tablespace ts_ij_l_pkey
  add datafile '\\.\ts_ij_l_pkey_100g3' size
9600m reuse;
alter tablespace ts_ij_l_pkey
  add datafile '\\.\ts_ij_l_pkey_100g4' size
9600m reuse;
alter tablespace ts_ij_l_pkey
  add datafile '\\.\ts_ij_l_pkey_100g5' size
9600m reuse;
alter tablespace ts_ij_l_pkey
  add datafile '\\.\ts_ij_l_pkey_100g2' size
9600m reuse;
}
# adding tpcd's ts_ij_l_rdate datafiles
*sql
{
alter tablespace ts_ij_l_rdate
  add datafile '\\.\ts_ij_l_rdate_100g4' size
8700m reuse;
alter tablespace ts_ij_l_rdate
  add datafile '\\.\ts_ij_l_rdate_100g5' size
8700m reuse;
alter tablespace ts_ij_l_rdate
  add datafile '\\.\ts_ij_l_rdate_100g2' size
8700m reuse;
alter tablespace ts_ij_l_rdate
  add datafile '\\.\ts_ij_l_rdate_100g3' size
8700m reuse;
}
# adding tpcd's ts_o datafiles
*sql
{
alter tablespace ts_o
  add datafile '\\.\ts_o_100g5' size 6000m reuse;
alter tablespace ts_o
  add datafile '\\.\ts_o_100g2' size 6000m reuse;
alter tablespace ts_o
  add datafile '\\.\ts_o_100g3' size 6000m reuse;
alter tablespace ts_o
  add datafile '\\.\ts_o_100g4' size 6000m reuse;
}
# adding tpcd's ts_ind_lored datafiles
*sql
{
alter tablespace ts_ind_lored
  add datafile '\\.\ts_ind_lored_100g2' size
4620m reuse;
alter tablespace ts_ind_lored
  add datafile '\\.\ts_ind_lored_100g3' size
4620m reuse;
alter tablespace ts_ind_lored
  add datafile '\\.\ts_ind_lored_100g4' size
4620m reuse;
alter tablespace ts_ind_lored
  add datafile '\\.\ts_ind_lored_100g5' size
4620m reuse;
}
# adding tpcd's ts_ij_l_lrid datafiles
*sql
{
alter tablespace ts_ij_l_lrid
  add datafile '\\.\ts_ij_l_lrid_100g3' size 4200m
reuse;
alter tablespace ts_ij_l_lrid
  add datafile '\\.\ts_ij_l_lrid_100g4' size 4200m
reuse;
alter tablespace ts_ij_l_lrid
  add datafile '\\.\ts_ij_l_lrid_100g5' size 4200m
reuse;
alter tablespace ts_ij_l_lrid
  add datafile '\\.\ts_ij_l_lrid_100g2' size 4200m
reuse;
}
# adding tpcd's ts_ij_l_orid datafiles
*sql
{
alter tablespace ts_ij_l_orid
  add datafile '\\.\ts_ij_orid_100g4' size 4200m
reuse;
alter tablespace ts_ij_l_orid
  add datafile '\\.\ts_ij_orid_100g5' size 4200m
reuse;
alter tablespace ts_ij_l_orid
  add datafile '\\.\ts_ij_orid_100g2' size 4200m
reuse;
alter tablespace ts_ij_l_orid
  add datafile '\\.\ts_ij_orid_100g3' size 4200m
reuse;
}
# adding tpcd's ts_mj_ppssrn datafiles
*sql
{
alter tablespace ts_mj_ppssrn
  add datafile '\\.\ts_mj_ppssrn_100g5' size
3863m reuse;
alter tablespace ts_mj_ppssrn
  add datafile '\\.\ts_mj_ppssrn_100g2' size
3863m reuse;
alter tablespace ts_mj_ppssrn
  add datafile '\\.\ts_mj_ppssrn_100g3' size
3863m reuse;
alter tablespace ts_mj_ppssrn
  add datafile '\\.\ts_mj_ppssrn_100g4' size
3863m reuse;
}
# adding tpcd's ts_p_cbp datafiles
*sql
{
alter tablespace ts_p_cbp
  add datafile '\\.\ts_p_cbp_100g2' size 3047m
reuse;
alter tablespace ts_p_cbp
  add datafile '\\.\ts_p_cbp_100g3' size 3047m
reuse;
alter tablespace ts_p_cbp
  add datafile '\\.\ts_p_cbp_100g4' size 3047m
reuse;
alter tablespace ts_p_cbp
  add datafile '\\.\ts_p_cbp_100g5' size 3047m
reuse;
}
# adding tpcd's ts_ps datafiles
*sql
{
alter tablespace ts_ps
  add datafile '\\.\ts_ps_100g3' size 2368m
reuse;
alter tablespace ts_ps
  add datafile '\\.\ts_ps_100g4' size 2368m
reuse;
alter tablespace ts_ps
  add datafile '\\.\ts_ps_100g5' size 2368m
reuse;
alter tablespace ts_ps
```

```
  add datafile '\\.\ts_ps_100g2' size 2368m
reuse;
}
# adding tpcd's ts_default datafiles
*sql
{
alter tablespace ts_default
  add datafile '\\.\ts_default_100g4' size 2000m
reuse;
alter tablespace ts_default
  add datafile '\\.\ts_default_100g5' size 2000m
reuse;
alter tablespace ts_default
  add datafile '\\.\ts_default_100g2' size 2000m
reuse;
alter tablespace ts_default
  add datafile '\\.\ts_default_100g3' size 2000m
reuse;
}
# adding tpcd's ts_ind_oclokod datafiles
*sql
{
alter tablespace ts_ind_oclokod
  add datafile '\\.\ts_ind_oclokod_100g5' size
1848m reuse;
alter tablespace ts_ind_oclokod
  add datafile '\\.\ts_ind_oclokod_100g2' size
1848m reuse;
alter tablespace ts_ind_oclokod
  add datafile '\\.\ts_ind_oclokod_100g3' size
1848m reuse;
alter tablespace ts_ind_oclokod
  add datafile '\\.\ts_ind_oclokod_100g4' size
1848m reuse;
}
# adding tpcd's ts_ij_p_ppsn datafiles
*sql
{
alter tablespace ts_ij_p_ppsn
  add datafile '\\.\ts_ij_p_ppsn_100g2' size
1560m reuse;
alter tablespace ts_ij_p_ppsn
  add datafile '\\.\ts_ij_p_ppsn_100g3' size
1560m reuse;
alter tablespace ts_ij_p_ppsn
  add datafile '\\.\ts_ij_p_ppsn_100g4' size
1560m reuse;
alter tablespace ts_ij_p_ppsn
  add datafile '\\.\ts_ij_p_ppsn_100g5' size
1560m reuse;
}
# adding tpcd's ts_ind_ookey datafiles
*sql
{
alter tablespace ts_ind_ookey
  add datafile '\\.\ts_ind_okey_100g3' size 1000m
reuse;
alter tablespace ts_ind_ookey
  add datafile '\\.\ts_ind_okey_100g4' size 1000m
reuse;
alter tablespace ts_ind_ookey
  add datafile '\\.\ts_ind_okey_100g5' size 1000m
reuse;
alter tablespace ts_ind_ookey
  add datafile '\\.\ts_ind_okey_100g2' size 1000m
reuse;
}
# adding tpcd's ts_constr datafiles
*sql
{
alter tablespace ts_constr
  add datafile '\\.\ts_constr_100g4' size 1000m
reuse;
alter tablespace ts_constr
  add datafile '\\.\ts_constr_100g5' size 1000m
reuse;
alter tablespace ts_constr
  add datafile '\\.\ts_constr_100g2' size 1000m
reuse;
alter tablespace ts_constr
  add datafile '\\.\ts_constr_100g3' size 1000m
reuse;
}
# adding tpcd's ts_ij_p_nnpsps datafiles
*sql
{
alter tablespace ts_ij_p_nnpsps
  add datafile '\\.\ts_ij_p_nnpsps_100g5' size
843m reuse;
alter tablespace ts_ij_p_nnpsps
  add datafile '\\.\ts_ij_p_nnpsps_100g2' size
843m reuse;
alter tablespace ts_ij_p_nnpsps
  add datafile '\\.\ts_ij_p_nnpsps_100g3' size
843m reuse;
alter tablespace ts_ij_p_nnpsps
  add datafile '\\.\ts_ij_p_nnpsps_100g4' size
843m reuse;
}
# adding tpcd's ts_p datafiles
*sql
{
alter tablespace ts_p
  add datafile '\\.\ts_p_100g2' size 605m reuse;
alter tablespace ts_p
  add datafile '\\.\ts_p_100g3' size 605m reuse;
alter tablespace ts_p
  add datafile '\\.\ts_p_100g4' size 605m reuse;
alter tablespace ts_p
  add datafile '\\.\ts_p_100g5' size 605m reuse;
}
# adding tpcd's ts_mllog datafiles
*sql
{
alter tablespace ts_mllog
  add datafile '\\.\ts_mllog_100g3' size 600m
reuse;
alter tablespace ts_mllog
  add datafile '\\.\ts_mllog_100g4' size 600m
reuse;
alter tablespace ts_mllog
  add datafile '\\.\ts_mllog_100g5' size 600m
reuse;
alter tablespace ts_mllog
  add datafile '\\.\ts_mllog_100g2' size 600m
reuse;
}
# adding tpcd's ts_molog datafiles
*sql
{
alter tablespace ts_molog
  add datafile '\\.\ts_molog_100g4' size 600m
reuse;
alter tablespace ts_molog
  add datafile '\\.\ts_molog_100g5' size 600m
reuse;
alter tablespace ts_molog
  add datafile '\\.\ts_molog_100g2' size 600m
```

```
reuse;
alter tablespace ts_molog
  add datafile '\\.\ts_molog_100g3' size 600m
reuse;
}
# adding tpcd's ts_maint datafiles
# adding tpcd's ts_c datafiles
*sql
{
alter tablespace ts_c
  add datafile '\\.\ts_c_100g5' size 511m reuse;
alter tablespace ts_c
  add datafile '\\.\ts_c_100g2' size 511m reuse;
alter tablespace ts_c
  add datafile '\\.\ts_c_100g3' size 511m reuse;
alter tablespace ts_c
  add datafile '\\.\ts_c_100g4' size 511m reuse;
}
# adding tpcd's ts_p_tp datafiles
*sql
{
alter tablespace ts_p_tp
  add datafile '\\.\ts_p_tp_100g2' size 458m
reuse;
alter tablespace ts_p_tp
  add datafile '\\.\ts_p_tp_100g3' size 458m
reuse;
alter tablespace ts_p_tp
  add datafile '\\.\ts_p_tp_100g4' size 458m
reuse;
alter tablespace ts_p_tp
  add datafile '\\.\ts_p_tp_100g5' size 458m
reuse;
}
# adding tpcd's ts_1 datafiles
*sql
{
alter tablespace ts_1
  add datafile '\\.\ts_13' size 200m reuse;
alter tablespace ts_1
  add datafile '\\.\ts_14' size 200m reuse;
alter tablespace ts_1
  add datafile '\\.\ts_15' size 200m reuse;
alter tablespace ts_1
  add datafile '\\.\ts_12' size 200m reuse;
}
# adding tpcd's ts_ij_l_nullrid datafiles
*sql
{
alter tablespace ts_ij_l_nullrid
  add datafile '\\.\ts_ij_nullrid_100g4' size 39m
reuse;
alter tablespace ts_ij_l_nullrid
  add datafile '\\.\ts_ij_nullrid_100g5' size 39m
reuse;
alter tablespace ts_ij_l_nullrid
  add datafile '\\.\ts_ij_nullrid_100g2' size 39m
reuse;
alter tablespace ts_ij_l_nullrid
  add datafile '\\.\ts_ij_nullrid_100g3' size 39m
reuse;
}
# adding tpcd's ts_s datafiles
*sql
{
alter tablespace ts_s
  add datafile '\\.\ts_s_100g5' size 31m reuse;
alter tablespace ts_s
  add datafile '\\.\ts_s_100g2' size 31m reuse;
alter tablespace ts_s
  add datafile '\\.\ts_s_100g3' size 31m reuse;
alter tablespace ts_s
  add datafile '\\.\ts_s_100g4' size 31m reuse;
}
# adding tpcd's ts_ma_lineitem datafiles
*sql
{
alter tablespace ts_ma_lineitem
  add datafile '\\.\ts_ma_lineitem_100g2' size 3m
reuse;
alter tablespace ts_ma_lineitem
  add datafile '\\.\ts_ma_lineitem_100g3' size 3m
reuse;
alter tablespace ts_ma_lineitem
  add datafile '\\.\ts_ma_lineitem_100g4' size 3m
reuse;
alter tablespace ts_ma_lineitem
  add datafile '\\.\ts_ma_lineitem_100g5' size 3m
reuse;
}
*sql
{
alter tablespace ts_maint
  add datafile '\\.\ts_maint_100g3' size 3m
reuse;
alter tablespace ts_maint
  add datafile '\\.\ts_maint_100g4' size 3m
reuse;
alter tablespace ts_maint
  add datafile '\\.\ts_maint_100g5' size 3m
reuse;
alter tablespace ts_maint
  add datafile '\\.\ts_maint_100g2' size 3m
reuse;
}
*wait
*bgoff
%e-sccre
```

100g_816_scuvo.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}
```

```
*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-scuvo
*bgon=1
################################################
############################
# Schema Creation Phase - Views ONLY (no
datafiles)
*time=Cycling database before phase 'scuvo'...
*time=Shutting down database...
*sql
{
shutdown normal;
}
*wait
*time=Starting database...
*sql
{
startup
pfile=e:\tpcd814_work\dout\dbs\hundgig\init_100g.
ora;
}
*wait
################################################
############################
# First I will create the materialized log for
the base tables
*sql
{
connect tpcd/tpcd;
drop materialized view log on lineitem;
create materialized view log on lineitem
tablespace ts_mllog
storage (initial 32k next 1m minextents 1
pctincrease 0)
parallel (degree 16)
nologging
partition by range (l_shipdate)
(
partition item1 values less than (to_date('1992-
03-01','YYYY-MM-DD'))
,
partition item2 values less than (to_date('1992-
03-24','YYYY-MM-DD'))
,
partition item3 values less than (to_date('1992-
04-14','YYYY-MM-DD'))
,
partition item4 values less than (to_date('1992-
05-01','YYYY-MM-DD'))
,
partition item5 values less than (to_date('1992-
05-16','YYYY-MM-DD'))
,
partition item6 values less than (to_date('1992-
06-01','YYYY-MM-DD'))
,
partition item7 values less than (to_date('1992-
06-16','YYYY-MM-DD'))
,
partition item8 values less than (to_date('1992-
07-01','YYYY-MM-DD'))
,
partition item9 values less than (to_date('1992-
07-16','YYYY-MM-DD'))
,
partition item10 values less than (to_date('1992-
08-01','YYYY-MM-DD'))
,
partition item11 values less than (to_date('1992-
08-16','YYYY-MM-DD'))
,
partition item12 values less than (to_date('1992-
09-01','YYYY-MM-DD'))
,
partition item13 values less than (to_date('1992-
09-16','YYYY-MM-DD'))
,
partition item14 values less than (to_date('1992-
10-01','YYYY-MM-DD'))
,
partition item15 values less than (to_date('1992-
10-16','YYYY-MM-DD'))
,
partition item16 values less than (to_date('1992-
11-01','YYYY-MM-DD'))
,
partition item17 values less than (to_date('1992-
11-16','YYYY-MM-DD'))
,
partition item18 values less than (to_date('1992-
12-01','YYYY-MM-DD'))
,
partition item19 values less than (to_date('1992-
12-16','YYYY-MM-DD'))
,
partition item20 values less than (to_date('1993-
01-01','YYYY-MM-DD'))
,
partition item21 values less than (to_date('1993-
01-16','YYYY-MM-DD'))
,
partition item22 values less than (to_date('1993-
02-01','YYYY-MM-DD'))
,
partition item23 values less than (to_date('1993-
02-16','YYYY-MM-DD'))
,
partition item24 values less than (to_date('1993-
03-01','YYYY-MM-DD'))
,
partition item25 values less than (to_date('1993-
03-16','YYYY-MM-DD'))
,
partition item26 values less than (to_date('1993-
04-01','YYYY-MM-DD'))
,
partition item27 values less than (to_date('1993-
04-16','YYYY-MM-DD'))
,
partition item28 values less than (to_date('1993-
05-01','YYYY-MM-DD'))
,
partition item29 values less than (to_date('1993-
05-16','YYYY-MM-DD'))
,
partition item30 values less than (to_date('1993-
06-01','YYYY-MM-DD'))
,
partition item31 values less than (to_date('1993-
06-16','YYYY-MM-DD'))
,
partition item32 values less than (to_date('1993-
```

```
07-01','YYYY-MM-DD'))
,
partition item33 values less than (to_date('1993-
07-16','YYYY-MM-DD'))
,
partition item34 values less than (to_date('1993-
08-01','YYYY-MM-DD'))
,
partition item35 values less than (to_date('1993-
08-16','YYYY-MM-DD'))
,
partition item36 values less than (to_date('1993-
09-01','YYYY-MM-DD'))
,
partition item37 values less than (to_date('1993-
09-16','YYYY-MM-DD'))
,
partition item38 values less than (to_date('1993-
10-01','YYYY-MM-DD'))
,
partition item39 values less than (to_date('1993-
10-16','YYYY-MM-DD'))
,
partition item40 values less than (to_date('1993-
11-01','YYYY-MM-DD'))
,
partition item41 values less than (to_date('1993-
11-16','YYYY-MM-DD'))
,
partition item42 values less than (to_date('1993-
12-01','YYYY-MM-DD'))
,
partition item43 values less than (to_date('1993-
12-16','YYYY-MM-DD'))
,
partition item44 values less than (to_date('1994-
01-01','YYYY-MM-DD'))
,
partition item45 values less than (to_date('1994-
01-16','YYYY-MM-DD'))
,
partition item46 values less than (to_date('1994-
02-01','YYYY-MM-DD'))
,
partition item47 values less than (to_date('1994-
02-16','YYYY-MM-DD'))
,
partition item48 values less than (to_date('1994-
03-01','YYYY-MM-DD'))
,
partition item49 values less than (to_date('1994-
03-16','YYYY-MM-DD'))
,
partition item50 values less than (to_date('1994-
04-01','YYYY-MM-DD'))
,
partition item51 values less than (to_date('1994-
04-16','YYYY-MM-DD'))
,
partition item52 values less than (to_date('1994-
05-01','YYYY-MM-DD'))
,
partition item53 values less than (to_date('1994-
05-16','YYYY-MM-DD'))
,
partition item54 values less than (to_date('1994-
06-01','YYYY-MM-DD'))
,
partition item55 values less than (to_date('1994-
06-16','YYYY-MM-DD'))
,
partition item56 values less than (to_date('1994-
07-01','YYYY-MM-DD'))
,
partition item57 values less than (to_date('1994-
07-16','YYYY-MM-DD'))
,
partition item58 values less than (to_date('1994-
08-01','YYYY-MM-DD'))
,
partition item59 values less than (to_date('1994-
08-16','YYYY-MM-DD'))
,
partition item60 values less than (to_date('1994-
09-01','YYYY-MM-DD'))
,
partition item61 values less than (to_date('1994-
09-16','YYYY-MM-DD'))
,
partition item62 values less than (to_date('1994-
10-01','YYYY-MM-DD'))
,
partition item63 values less than (to_date('1994-
10-16','YYYY-MM-DD'))
,
partition item64 values less than (to_date('1994-
11-01','YYYY-MM-DD'))
,
partition item65 values less than (to_date('1994-
11-16','YYYY-MM-DD'))
,
partition item66 values less than (to_date('1994-
12-01','YYYY-MM-DD'))
,
partition item67 values less than (to_date('1994-
12-16','YYYY-MM-DD'))
,
partition item68 values less than (to_date('1995-
01-01','YYYY-MM-DD'))
,
partition item69 values less than (to_date('1995-
01-16','YYYY-MM-DD'))
,
partition item70 values less than (to_date('1995-
02-01','YYYY-MM-DD'))
,
partition item71 values less than (to_date('1995-
02-16','YYYY-MM-DD'))
,
partition item72 values less than (to_date('1995-
03-01','YYYY-MM-DD'))
,
partition item73 values less than (to_date('1995-
03-16','YYYY-MM-DD'))
,
partition item74 values less than (to_date('1995-
04-01','YYYY-MM-DD'))
,
partition item75 values less than (to_date('1995-
04-16','YYYY-MM-DD'))
,
partition item76 values less than (to_date('1995-
05-01','YYYY-MM-DD'))
,
partition item77 values less than (to_date('1995-
05-16','YYYY-MM-DD'))
,
partition item78 values less than (to_date('1995-
```

```
06-01','YYYY-MM-DD'))
,
partition item79 values less than (to_date('1995-
06-16','YYYY-MM-DD'))
,
partition item80 values less than (to_date('1995-
07-01','YYYY-MM-DD'))
,
partition item81 values less than (to_date('1995-
07-16','YYYY-MM-DD'))
,
partition item82 values less than (to_date('1995-
08-01','YYYY-MM-DD'))
,
partition item83 values less than (to_date('1995-
08-16','YYYY-MM-DD'))
,
partition item84 values less than (to_date('1995-
09-01','YYYY-MM-DD'))
,
partition item85 values less than (to_date('1995-
09-16','YYYY-MM-DD'))
,
partition item86 values less than (to_date('1995-
10-01','YYYY-MM-DD'))
,
partition item87 values less than (to_date('1995-
10-16','YYYY-MM-DD'))
,
partition item88 values less than (to_date('1995-
11-01','YYYY-MM-DD'))
,
partition item89 values less than (to_date('1995-
11-16','YYYY-MM-DD'))
,
partition item90 values less than (to_date('1995-
12-01','YYYY-MM-DD'))
,
partition item91 values less than (to_date('1995-
12-16','YYYY-MM-DD'))
,
partition item92 values less than (to_date('1996-
01-01','YYYY-MM-DD'))
,
partition item93 values less than (to_date('1996-
01-16','YYYY-MM-DD'))
,
partition item94 values less than (to_date('1996-
02-01','YYYY-MM-DD'))
,
partition item95 values less than (to_date('1996-
02-16','YYYY-MM-DD'))
,
partition item96 values less than (to_date('1996-
03-01','YYYY-MM-DD'))
,
partition item97 values less than (to_date('1996-
03-16','YYYY-MM-DD'))
,
partition item98 values less than (to_date('1996-
04-01','YYYY-MM-DD'))
,
partition item99 values less than (to_date('1996-
04-16','YYYY-MM-DD'))
,
partition item100 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
,
partition item101 values less than
(to_date('1996-05-16','YYYY-MM-DD'))
,
partition item102 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
,
partition item103 values less than
(to_date('1996-06-16','YYYY-MM-DD'))
,
partition item104 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
,
partition item105 values less than
(to_date('1996-07-16','YYYY-MM-DD'))
,
partition item106 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
,
partition item107 values less than
(to_date('1996-08-16','YYYY-MM-DD'))
,
partition item108 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
,
partition item109 values less than
(to_date('1996-09-16','YYYY-MM-DD'))
,
partition item110 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
,
partition item111 values less than
(to_date('1996-10-16','YYYY-MM-DD'))
,
partition item112 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
,
partition item113 values less than
(to_date('1996-11-16','YYYY-MM-DD'))
,
partition item114 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
,
partition item115 values less than
(to_date('1996-12-16','YYYY-MM-DD'))
,
partition item116 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
,
partition item117 values less than
(to_date('1997-01-16','YYYY-MM-DD'))
,
partition item118 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
,
partition item119 values less than
(to_date('1997-02-16','YYYY-MM-DD'))
,
partition item120 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
,
partition item121 values less than
(to_date('1997-03-16','YYYY-MM-DD'))
,
partition item122 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
,
partition item123 values less than
(to_date('1997-04-16','YYYY-MM-DD'))
,
partition item124 values less than
```

```
(to_date('1997-05-01','YYYY-MM-DD'))
,
partition item125 values less than
(to_date('1997-05-16','YYYY-MM-DD'))
,
partition item126 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
,
partition item127 values less than
(to_date('1997-06-16','YYYY-MM-DD'))
,
partition item128 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
,
partition item129 values less than
(to_date('1997-07-16','YYYY-MM-DD'))
,
partition item130 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
,
partition item131 values less than
(to_date('1997-08-16','YYYY-MM-DD'))
,
partition item132 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
,
partition item133 values less than
(to_date('1997-09-16','YYYY-MM-DD'))
,
partition item134 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
,
partition item135 values less than
(to_date('1997-10-16','YYYY-MM-DD'))
,
partition item136 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
,
partition item137 values less than
(to_date('1997-11-16','YYYY-MM-DD'))
,
partition item138 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
,
partition item139 values less than
(to_date('1997-12-16','YYYY-MM-DD'))
,
partition item140 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
,
partition item141 values less than
(to_date('1998-01-16','YYYY-MM-DD'))
,
partition item142 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
,
partition item143 values less than
(to_date('1998-02-16','YYYY-MM-DD'))
,
partition item144 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
,
partition item145 values less than
(to_date('1998-03-16','YYYY-MM-DD'))
,
partition item146 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
,
partition item147 values less than
(to_date('1998-04-16','YYYY-MM-DD'))
,
partition item148 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
,
partition item149 values less than
(to_date('1998-05-16','YYYY-MM-DD'))
,
partition item150 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
,
partition item151 values less than
(to_date('1998-06-16','YYYY-MM-DD'))
,
partition item152 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
,
partition item153 values less than
(to_date('1998-07-16','YYYY-MM-DD'))
,
partition item154 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
,
partition item155 values less than
(to_date('1998-08-16','YYYY-MM-DD'))
,
partition item156 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
,
partition item157 values less than
(to_date('1998-09-16','YYYY-MM-DD'))
,
partition item158 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
,
partition item159 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
,
partition item160 values less than (MAXVALUE)
)
with rowid (
    l_shipdate            ,
    l_returnflag          ,
    l_linestatus          ,
    l_quantity            ,
    l_extendedprice       ,
    l_discount            ,
    l_tax                 ,
     l_suppkey
) including new values
;
drop materialized view log on orders;
create materialized view log on orders
tablespace ts_default
storage (initial 32k next 512k minextents 1
pctincrease 0)
parallel (degree 16)
nologging
partition by hash (o_orderkey)
partitions 160
with rowid (
  o_orderkey
) including new values
;
drop materialized view log on customer;
create materialized view log on customer
tablespace ts_default
with rowid including new values
;
```

```
drop materialized view log on supplier;
create materialized view log on supplier
tablespace ts_default
with rowid including new values;
drop materialized view log on parts;
create materialized view log on parts
tablespace ts_default
with rowid including new values;
drop materialized view log on partsupp;
create materialized view log on partsupp
tablespace ts_default
with rowid including new values;
drop materialized view log on nation;
create materialized view log on nation
tablespace ts_default
with rowid including new values;
drop materialized view log on region;
create materialized view log on region
tablespace ts_default
with rowid including new values;
}
*wait
################################################
############################
# Now I can create the views with the
corresponding materialized logs
*sql
{
connect tpcd/tpcd;
drop materialized view ma_lineitem;
create materialized view ma_lineitem
pctfree 2
tablespace ts_ma_lineitem
storage (initial 512k next 512k maxextents
unlimited pctincrease 0)
parallel (degree 1)
nologging
BUILD immediate
REFRESH fast
ON commit
enable query rewrite
as select
  l_shipdate,
  l_returnflag,
  l_linestatus,
  sum(l_quantity) as sum_qty,
  count(l_quantity) as count_qty,
  sum(l_extendedprice) as sum_base_price,
  count(l_extendedprice) as count_price,
  sum(l_extendedprice * (1 - l_discount)) as
sum_disc_price,
  count(l_extendedprice * (1 - l_discount)) as
count_disc_price,
  sum(l_extendedprice * (1 - l_discount) * (1 +
l_tax)) as sum_charge,
  count(l_extendedprice * (1 - l_discount) * (1 +
l_tax)) as count_charge,
  sum(l_discount) as sum_disc,
  count(l_discount) as count_disc,
  count(*) as count_order
FROM
  lineitem
GROUP BY
  l_returnflag,
  l_linestatus,
  l_shipdate
;
}
*sql
{
connect tpcd/tpcd;
drop materialized view mj_locs;
create materialized view mj_locs
pctfree 2
tablespace ts_mj_locs
storage (initial 20m next 20m pctincrease 0
maxextents unlimited)
parallel (degree 20)
nologging
partition by range (l_shipdate)
subpartition by hash(c_mktsegment)
subpartitions 5
(
partition mj_locs1 values less than
(to_date('1992-03-01','YYYY-MM-DD'))
,
partition mj_locs2 values less than
(to_date('1992-03-24','YYYY-MM-DD'))
,
partition mj_locs3 values less than
(to_date('1992-04-14','YYYY-MM-DD'))
,
partition mj_locs4 values less than
(to_date('1992-05-01','YYYY-MM-DD'))
,
partition mj_locs5 values less than
(to_date('1992-05-16','YYYY-MM-DD'))
,
partition mj_locs6 values less than
(to_date('1992-06-01','YYYY-MM-DD'))
,
partition mj_locs7 values less than
(to_date('1992-06-16','YYYY-MM-DD'))
,
partition mj_locs8 values less than
(to_date('1992-07-01','YYYY-MM-DD'))
,
partition mj_locs9 values less than
(to_date('1992-07-16','YYYY-MM-DD'))
,
partition mj_locs10 values less than
(to_date('1992-08-01','YYYY-MM-DD'))
,
partition mj_locs11 values less than
(to_date('1992-08-16','YYYY-MM-DD'))
,
partition mj_locs12 values less than
(to_date('1992-09-01','YYYY-MM-DD'))
,
partition mj_locs13 values less than
(to_date('1992-09-16','YYYY-MM-DD'))
,
partition mj_locs14 values less than
(to_date('1992-10-01','YYYY-MM-DD'))
,
partition mj_locs15 values less than
(to_date('1992-10-16','YYYY-MM-DD'))
,
partition mj_locs16 values less than
(to_date('1992-11-01','YYYY-MM-DD'))
,
partition mj_locs17 values less than
(to_date('1992-11-16','YYYY-MM-DD'))
,
partition mj_locs18 values less than
(to_date('1992-12-01','YYYY-MM-DD'))
,
partition mj_locs19 values less than
```

```
(to_date('1992-12-16','YYYY-MM-DD'))
,
partition mj_locs20 values less than
(to_date('1993-01-01','YYYY-MM-DD'))
,
partition mj_locs21 values less than
(to_date('1993-01-16','YYYY-MM-DD'))
,
partition mj_locs22 values less than
(to_date('1993-02-01','YYYY-MM-DD'))
,
partition mj_locs23 values less than
(to_date('1993-02-16','YYYY-MM-DD'))
,
partition mj_locs24 values less than
(to_date('1993-03-01','YYYY-MM-DD'))
,
partition mj_locs25 values less than
(to_date('1993-03-16','YYYY-MM-DD'))
,
partition mj_locs26 values less than
(to_date('1993-04-01','YYYY-MM-DD'))
,
partition mj_locs27 values less than
(to_date('1993-04-16','YYYY-MM-DD'))
,
partition mj_locs28 values less than
(to_date('1993-05-01','YYYY-MM-DD'))
,
partition mj_locs29 values less than
(to_date('1993-05-16','YYYY-MM-DD'))
,
partition mj_locs30 values less than
(to_date('1993-06-01','YYYY-MM-DD'))
,
partition mj_locs31 values less than
(to_date('1993-06-16','YYYY-MM-DD'))
,
partition mj_locs32 values less than
(to_date('1993-07-01','YYYY-MM-DD'))
,
partition mj_locs33 values less than
(to_date('1993-07-16','YYYY-MM-DD'))
,
partition mj_locs34 values less than
(to_date('1993-08-01','YYYY-MM-DD'))
,
partition mj_locs35 values less than
(to_date('1993-08-16','YYYY-MM-DD'))
,
partition mj_locs36 values less than
(to_date('1993-09-01','YYYY-MM-DD'))
,
partition mj_locs37 values less than
(to_date('1993-09-16','YYYY-MM-DD'))
,
partition mj_locs38 values less than
(to_date('1993-10-01','YYYY-MM-DD'))
,
partition mj_locs39 values less than
(to_date('1993-10-16','YYYY-MM-DD'))
,
partition mj_locs40 values less than
(to_date('1993-11-01','YYYY-MM-DD'))
,
partition mj_locs41 values less than
(to_date('1993-11-16','YYYY-MM-DD'))
,
partition mj_locs42 values less than
(to_date('1993-12-01','YYYY-MM-DD'))
,
partition mj_locs43 values less than
(to_date('1993-12-16','YYYY-MM-DD'))
,
partition mj_locs44 values less than
(to_date('1994-01-01','YYYY-MM-DD'))
,
partition mj_locs45 values less than
(to_date('1994-01-16','YYYY-MM-DD'))
,
partition mj_locs46 values less than
(to_date('1994-02-01','YYYY-MM-DD'))
,
partition mj_locs47 values less than
(to_date('1994-02-16','YYYY-MM-DD'))
,
partition mj_locs48 values less than
(to_date('1994-03-01','YYYY-MM-DD'))
,
partition mj_locs49 values less than
(to_date('1994-03-16','YYYY-MM-DD'))
,
partition mj_locs50 values less than
(to_date('1994-04-01','YYYY-MM-DD'))
,
partition mj_locs51 values less than
(to_date('1994-04-16','YYYY-MM-DD'))
,
partition mj_locs52 values less than
(to_date('1994-05-01','YYYY-MM-DD'))
,
partition mj_locs53 values less than
(to_date('1994-05-16','YYYY-MM-DD'))
,
partition mj_locs54 values less than
(to_date('1994-06-01','YYYY-MM-DD'))
,
partition mj_locs55 values less than
(to_date('1994-06-16','YYYY-MM-DD'))
,
partition mj_locs56 values less than
(to_date('1994-07-01','YYYY-MM-DD'))
,
partition mj_locs57 values less than
(to_date('1994-07-16','YYYY-MM-DD'))
,
partition mj_locs58 values less than
(to_date('1994-08-01','YYYY-MM-DD'))
,
partition mj_locs59 values less than
(to_date('1994-08-16','YYYY-MM-DD'))
,
partition mj_locs60 values less than
(to_date('1994-09-01','YYYY-MM-DD'))
,
partition mj_locs61 values less than
(to_date('1994-09-16','YYYY-MM-DD'))
,
partition mj_locs62 values less than
(to_date('1994-10-01','YYYY-MM-DD'))
,
partition mj_locs63 values less than
(to_date('1994-10-16','YYYY-MM-DD'))
,
partition mj_locs64 values less than
(to_date('1994-11-01','YYYY-MM-DD'))
,
partition mj_locs65 values less than
```

```
(to_date('1994-11-16','YYYY-MM-DD'))
,
partition mj_locs66 values less than
(to_date('1994-12-01','YYYY-MM-DD'))
,
partition mj_locs67 values less than
(to_date('1994-12-16','YYYY-MM-DD'))
,
partition mj_locs68 values less than
(to_date('1995-01-01','YYYY-MM-DD'))
,
partition mj_locs69 values less than
(to_date('1995-01-16','YYYY-MM-DD'))
,
partition mj_locs70 values less than
(to_date('1995-02-01','YYYY-MM-DD'))
,
partition mj_locs71 values less than
(to_date('1995-02-16','YYYY-MM-DD'))
,
partition mj_locs72 values less than
(to_date('1995-03-01','YYYY-MM-DD'))
,
partition mj_locs73 values less than
(to_date('1995-03-16','YYYY-MM-DD'))
,
partition mj_locs74 values less than
(to_date('1995-04-01','YYYY-MM-DD'))
,
partition mj_locs75 values less than
(to_date('1995-04-16','YYYY-MM-DD'))
,
partition mj_locs76 values less than
(to_date('1995-05-01','YYYY-MM-DD'))
,
partition mj_locs77 values less than
(to_date('1995-05-16','YYYY-MM-DD'))
,
partition mj_locs78 values less than
(to_date('1995-06-01','YYYY-MM-DD'))
,
partition mj_locs79 values less than
(to_date('1995-06-16','YYYY-MM-DD'))
,
partition mj_locs80 values less than
(to_date('1995-07-01','YYYY-MM-DD'))
,
partition mj_locs81 values less than
(to_date('1995-07-16','YYYY-MM-DD'))
,
partition mj_locs82 values less than
(to_date('1995-08-01','YYYY-MM-DD'))
,
partition mj_locs83 values less than
(to_date('1995-08-16','YYYY-MM-DD'))
,
partition mj_locs84 values less than
(to_date('1995-09-01','YYYY-MM-DD'))
,
partition mj_locs85 values less than
(to_date('1995-09-16','YYYY-MM-DD'))
,
partition mj_locs86 values less than
(to_date('1995-10-01','YYYY-MM-DD'))
,
partition mj_locs87 values less than
(to_date('1995-10-16','YYYY-MM-DD'))
,
partition mj_locs88 values less than
(to_date('1995-11-01','YYYY-MM-DD'))
,
partition mj_locs89 values less than
(to_date('1995-11-16','YYYY-MM-DD'))
,
partition mj_locs90 values less than
(to_date('1995-12-01','YYYY-MM-DD'))
,
partition mj_locs91 values less than
(to_date('1995-12-16','YYYY-MM-DD'))
,
partition mj_locs92 values less than
(to_date('1996-01-01','YYYY-MM-DD'))
,
partition mj_locs93 values less than
(to_date('1996-01-16','YYYY-MM-DD'))
,
partition mj_locs94 values less than
(to_date('1996-02-01','YYYY-MM-DD'))
,
partition mj_locs95 values less than
(to_date('1996-02-16','YYYY-MM-DD'))
,
partition mj_locs96 values less than
(to_date('1996-03-01','YYYY-MM-DD'))
,
partition mj_locs97 values less than
(to_date('1996-03-16','YYYY-MM-DD'))
,
partition mj_locs98 values less than
(to_date('1996-04-01','YYYY-MM-DD'))
,
partition mj_locs99 values less than
(to_date('1996-04-16','YYYY-MM-DD'))
,
partition mj_locs100 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
,
partition mj_locs101 values less than
(to_date('1996-05-16','YYYY-MM-DD'))
,
partition mj_locs102 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
,
partition mj_locs103 values less than
(to_date('1996-06-16','YYYY-MM-DD'))
,
partition mj_locs104 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
,
partition mj_locs105 values less than
(to_date('1996-07-16','YYYY-MM-DD'))
,
partition mj_locs106 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
,
partition mj_locs107 values less than
(to_date('1996-08-16','YYYY-MM-DD'))
,
partition mj_locs108 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
,
partition mj_locs109 values less than
(to_date('1996-09-16','YYYY-MM-DD'))
,
partition mj_locs110 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
,
partition mj_locs111 values less than
```

```
(to_date('1996-10-16','YYYY-MM-DD'))
,
partition mj_locs112 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
,
partition mj_locs113 values less than
(to_date('1996-11-16','YYYY-MM-DD'))
,
partition mj_locs114 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
,
partition mj_locs115 values less than
(to_date('1996-12-16','YYYY-MM-DD'))
,
partition mj_locs116 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
,
partition mj_locs117 values less than
(to_date('1997-01-16','YYYY-MM-DD'))
,
partition mj_locs118 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
,
partition mj_locs119 values less than
(to_date('1997-02-16','YYYY-MM-DD'))
,
partition mj_locs120 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
,
partition mj_locs121 values less than
(to_date('1997-03-16','YYYY-MM-DD'))
,
partition mj_locs122 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
,
partition mj_locs123 values less than
(to_date('1997-04-16','YYYY-MM-DD'))
,
partition mj_locs124 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
,
partition mj_locs125 values less than
(to_date('1997-05-16','YYYY-MM-DD'))
,
partition mj_locs126 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
,
partition mj_locs127 values less than
(to_date('1997-06-16','YYYY-MM-DD'))
,
partition mj_locs128 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
,
partition mj_locs129 values less than
(to_date('1997-07-16','YYYY-MM-DD'))
,
partition mj_locs130 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
,
partition mj_locs131 values less than
(to_date('1997-08-16','YYYY-MM-DD'))
,
partition mj_locs132 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
,
partition mj_locs133 values less than
(to_date('1997-09-16','YYYY-MM-DD'))
,
partition mj_locs134 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
,
partition mj_locs135 values less than
(to_date('1997-10-16','YYYY-MM-DD'))
,
partition mj_locs136 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
,
partition mj_locs137 values less than
(to_date('1997-11-16','YYYY-MM-DD'))
,
partition mj_locs138 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
,
partition mj_locs139 values less than
(to_date('1997-12-16','YYYY-MM-DD'))
,
partition mj_locs140 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
,
partition mj_locs141 values less than
(to_date('1998-01-16','YYYY-MM-DD'))
,
partition mj_locs142 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
,
partition mj_locs143 values less than
(to_date('1998-02-16','YYYY-MM-DD'))
,
partition mj_locs144 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
,
partition mj_locs145 values less than
(to_date('1998-03-16','YYYY-MM-DD'))
,
partition mj_locs146 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
,
partition mj_locs147 values less than
(to_date('1998-04-16','YYYY-MM-DD'))
,
partition mj_locs148 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
,
partition mj_locs149 values less than
(to_date('1998-05-16','YYYY-MM-DD'))
,
partition mj_locs150 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
,
partition mj_locs151 values less than
(to_date('1998-06-16','YYYY-MM-DD'))
,
partition mj_locs152 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
,
partition mj_locs153 values less than
(to_date('1998-07-16','YYYY-MM-DD'))
,
partition mj_locs154 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
,
partition mj_locs155 values less than
(to_date('1998-08-16','YYYY-MM-DD'))
,
partition mj_locs156 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
,
partition mj_locs157 values less than
```

```
(to_date('1998-09-16','YYYY-MM-DD'))
,
partition mj_locs158 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
,
partition mj_locs159 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
,
partition mj_locs160 values less than (MAXVALUE)
)
enable row movement
BUILD immediate
REFRESH fast
ON commit
enable query rewrite
as select
/*+ ordered user_hash (customer, orders,
lineitem, supplier) */
  l_shipdate,
  o_orderdate,
  l_extendedprice,
  l_discount,
  c_custkey,
  s_suppkey,
  c_nationkey,
  s_nationkey,
  ajm(orders.rowid) as o_ajm,
  ajm(customer.rowid) as c_ajm,
  ajm(supplier.rowid) as s_ajm,
  o_orderkey,
  l_receiptdate,
  l_commitdate,
  l_quantity,
  l_returnflag,
  l_partkey,
  l_suppkey,
  o_orderpriority,
  o_shippriority,
  o_custkey,
  c_mktsegment,
  l_orderkey,
  l_shipmode,
  lineitem.rowid as l_rowid,
  orders.rowid as o_rowid,
  customer.rowid as c_rowid,
  supplier.rowid as s_rowid
FROM
 lineitem,
 supplier,
 orders,
 customer
WHERE
 l_orderkey=o_orderkey(+)
 AND o_custkey=c_custkey(+)
 AND l_suppkey=s_suppkey(+)
order by o_orderkey
;
}
*sql
{
connect tpcd/tpcd;
drop materialized view ma_l_sdfy;
create materialized view ma_l_sdfy
pctfree 2
tablespace ts_ma_lineitem
storage (initial 520k next 520k maxextents
unlimited pctincrease 0)
parallel (degree 1)
nologging
BUILD immediate
REFRESH fast
ON commit
enable query rewrite
as select
  l_discount ,
  l_quantity ,
  dfy(l_shipdate) as dfy ,
  sum(l_extendedprice * l_discount) as volume ,
  count(l_extendedprice * l_discount) as
count_volume ,
  count(*) as count_grp
from
  lineitem
group by
  l_discount ,
  l_quantity ,
  dfy(l_shipdate)
;
}
*sql
{
connect tpcd/tpcd;
drop materialized view log on mj_locs;
create materialized view log on mj_locs
storage (initial 32k next 1m minextents 1
pctincrease 0)
parallel (degree 20)
nologging
partition by range (l_shipdate)
(
partition item1 values less than (to_date('1992-
03-01','YYYY-MM-DD'))
,
partition item2 values less than (to_date('1992-
03-24','YYYY-MM-DD'))
,
partition item3 values less than (to_date('1992-
04-14','YYYY-MM-DD'))
,
partition item4 values less than (to_date('1992-
05-01','YYYY-MM-DD'))
,
partition item5 values less than (to_date('1992-
05-16','YYYY-MM-DD'))
,
partition item6 values less than (to_date('1992-
06-01','YYYY-MM-DD'))
,
partition item7 values less than (to_date('1992-
06-16','YYYY-MM-DD'))
,
partition item8 values less than (to_date('1992-
07-01','YYYY-MM-DD'))
,
partition item9 values less than (to_date('1992-
07-16','YYYY-MM-DD'))
,
partition item10 values less than (to_date('1992-
08-01','YYYY-MM-DD'))
,
partition item11 values less than (to_date('1992-
08-16','YYYY-MM-DD'))
,
partition item12 values less than (to_date('1992-
09-01','YYYY-MM-DD'))
,
partition item13 values less than (to_date('1992-
09-16','YYYY-MM-DD'))
```

```
,
partition item14 values less than (to_date('1992-
10-01','YYYY-MM-DD'))
,
partition item15 values less than (to_date('1992-
10-16','YYYY-MM-DD'))
,
partition item16 values less than (to_date('1992-
11-01','YYYY-MM-DD'))
,
partition item17 values less than (to_date('1992-
11-16','YYYY-MM-DD'))
,
partition item18 values less than (to_date('1992-
12-01','YYYY-MM-DD'))
,
partition item19 values less than (to_date('1992-
12-16','YYYY-MM-DD'))
,
partition item20 values less than (to_date('1993-
01-01','YYYY-MM-DD'))
,
partition item21 values less than (to_date('1993-
01-16','YYYY-MM-DD'))
,
partition item22 values less than (to_date('1993-
02-01','YYYY-MM-DD'))
,
partition item23 values less than (to_date('1993-
02-16','YYYY-MM-DD'))
,
partition item24 values less than (to_date('1993-
03-01','YYYY-MM-DD'))
,
partition item25 values less than (to_date('1993-
03-16','YYYY-MM-DD'))
,
partition item26 values less than (to_date('1993-
04-01','YYYY-MM-DD'))
,
partition item27 values less than (to_date('1993-
04-16','YYYY-MM-DD'))
,
partition item28 values less than (to_date('1993-
05-01','YYYY-MM-DD'))
,
partition item29 values less than (to_date('1993-
05-16','YYYY-MM-DD'))
,
partition item30 values less than (to_date('1993-
06-01','YYYY-MM-DD'))
,
partition item31 values less than (to_date('1993-
06-16','YYYY-MM-DD'))
,
partition item32 values less than (to_date('1993-
07-01','YYYY-MM-DD'))
,
partition item33 values less than (to_date('1993-
07-16','YYYY-MM-DD'))
,
partition item34 values less than (to_date('1993-
08-01','YYYY-MM-DD'))
,
partition item35 values less than (to_date('1993-
08-16','YYYY-MM-DD'))
,
partition item36 values less than (to_date('1993-
09-01','YYYY-MM-DD'))
,
partition item37 values less than (to_date('1993-
09-16','YYYY-MM-DD'))
,
partition item38 values less than (to_date('1993-
10-01','YYYY-MM-DD'))
,
partition item39 values less than (to_date('1993-
10-16','YYYY-MM-DD'))
,
partition item40 values less than (to_date('1993-
11-01','YYYY-MM-DD'))
,
partition item41 values less than (to_date('1993-
11-16','YYYY-MM-DD'))
,
partition item42 values less than (to_date('1993-
12-01','YYYY-MM-DD'))
,
partition item43 values less than (to_date('1993-
12-16','YYYY-MM-DD'))
,
partition item44 values less than (to_date('1994-
01-01','YYYY-MM-DD'))
,
partition item45 values less than (to_date('1994-
01-16','YYYY-MM-DD'))
,
partition item46 values less than (to_date('1994-
02-01','YYYY-MM-DD'))
,
partition item47 values less than (to_date('1994-
02-16','YYYY-MM-DD'))
,
partition item48 values less than (to_date('1994-
03-01','YYYY-MM-DD'))
,
partition item49 values less than (to_date('1994-
03-16','YYYY-MM-DD'))
,
partition item50 values less than (to_date('1994-
04-01','YYYY-MM-DD'))
,
partition item51 values less than (to_date('1994-
04-16','YYYY-MM-DD'))
,
partition item52 values less than (to_date('1994-
05-01','YYYY-MM-DD'))
,
partition item53 values less than (to_date('1994-
05-16','YYYY-MM-DD'))
,
partition item54 values less than (to_date('1994-
06-01','YYYY-MM-DD'))
,
partition item55 values less than (to_date('1994-
06-16','YYYY-MM-DD'))
,
partition item56 values less than (to_date('1994-
07-01','YYYY-MM-DD'))
,
partition item57 values less than (to_date('1994-
07-16','YYYY-MM-DD'))
,
partition item58 values less than (to_date('1994-
08-01','YYYY-MM-DD'))
,
partition item59 values less than (to_date('1994-
08-16','YYYY-MM-DD'))
```

```
,
partition item60 values less than (to_date('1994-
09-01','YYYY-MM-DD'))
,
partition item61 values less than (to_date('1994-
09-16','YYYY-MM-DD'))
,
partition item62 values less than (to_date('1994-
10-01','YYYY-MM-DD'))
,
partition item63 values less than (to_date('1994-
10-16','YYYY-MM-DD'))
,
partition item64 values less than (to_date('1994-
11-01','YYYY-MM-DD'))
,
partition item65 values less than (to_date('1994-
11-16','YYYY-MM-DD'))
,
partition item66 values less than (to_date('1994-
12-01','YYYY-MM-DD'))
,
partition item67 values less than (to_date('1994-
12-16','YYYY-MM-DD'))
,
partition item68 values less than (to_date('1995-
01-01','YYYY-MM-DD'))
,
partition item69 values less than (to_date('1995-
01-16','YYYY-MM-DD'))
,
partition item70 values less than (to_date('1995-
02-01','YYYY-MM-DD'))
,
partition item71 values less than (to_date('1995-
02-16','YYYY-MM-DD'))
,
partition item72 values less than (to_date('1995-
03-01','YYYY-MM-DD'))
,
partition item73 values less than (to_date('1995-
03-16','YYYY-MM-DD'))
,
partition item74 values less than (to_date('1995-
04-01','YYYY-MM-DD'))
,
partition item75 values less than (to_date('1995-
04-16','YYYY-MM-DD'))
,
partition item76 values less than (to_date('1995-
05-01','YYYY-MM-DD'))
,
partition item77 values less than (to_date('1995-
05-16','YYYY-MM-DD'))
,
partition item78 values less than (to_date('1995-
06-01','YYYY-MM-DD'))
,
partition item79 values less than (to_date('1995-
06-16','YYYY-MM-DD'))
,
partition item80 values less than (to_date('1995-
07-01','YYYY-MM-DD'))
,
partition item81 values less than (to_date('1995-
07-16','YYYY-MM-DD'))
,
partition item82 values less than (to_date('1995-
08-01','YYYY-MM-DD'))
,
partition item83 values less than (to_date('1995-
08-16','YYYY-MM-DD'))
,
partition item84 values less than (to_date('1995-
09-01','YYYY-MM-DD'))
,
partition item85 values less than (to_date('1995-
09-16','YYYY-MM-DD'))
,
partition item86 values less than (to_date('1995-
10-01','YYYY-MM-DD'))
,
partition item87 values less than (to_date('1995-
10-16','YYYY-MM-DD'))
,
partition item88 values less than (to_date('1995-
11-01','YYYY-MM-DD'))
,
partition item89 values less than (to_date('1995-
11-16','YYYY-MM-DD'))
,
partition item90 values less than (to_date('1995-
12-01','YYYY-MM-DD'))
,
partition item91 values less than (to_date('1995-
12-16','YYYY-MM-DD'))
,
partition item92 values less than (to_date('1996-
01-01','YYYY-MM-DD'))
,
partition item93 values less than (to_date('1996-
01-16','YYYY-MM-DD'))
,
partition item94 values less than (to_date('1996-
02-01','YYYY-MM-DD'))
,
partition item95 values less than (to_date('1996-
02-16','YYYY-MM-DD'))
,
partition item96 values less than (to_date('1996-
03-01','YYYY-MM-DD'))
,
partition item97 values less than (to_date('1996-
03-16','YYYY-MM-DD'))
,
partition item98 values less than (to_date('1996-
04-01','YYYY-MM-DD'))
,
partition item99 values less than (to_date('1996-
04-16','YYYY-MM-DD'))
,
partition item100 values less than
(to_date('1996-05-01','YYYY-MM-DD'))
,
partition item101 values less than
(to_date('1996-05-16','YYYY-MM-DD'))
,
partition item102 values less than
(to_date('1996-06-01','YYYY-MM-DD'))
,
partition item103 values less than
(to_date('1996-06-16','YYYY-MM-DD'))
,
partition item104 values less than
(to_date('1996-07-01','YYYY-MM-DD'))
,
partition item105 values less than
(to_date('1996-07-16','YYYY-MM-DD'))
```

```
,
partition item106 values less than
(to_date('1996-08-01','YYYY-MM-DD'))
,
partition item107 values less than
(to_date('1996-08-16','YYYY-MM-DD'))
,
partition item108 values less than
(to_date('1996-09-01','YYYY-MM-DD'))
,
partition item109 values less than
(to_date('1996-09-16','YYYY-MM-DD'))
,
partition item110 values less than
(to_date('1996-10-01','YYYY-MM-DD'))
,
partition item111 values less than
(to_date('1996-10-16','YYYY-MM-DD'))
,
partition item112 values less than
(to_date('1996-11-01','YYYY-MM-DD'))
,
partition item113 values less than
(to_date('1996-11-16','YYYY-MM-DD'))
,
partition item114 values less than
(to_date('1996-12-01','YYYY-MM-DD'))
,
partition item115 values less than
(to_date('1996-12-16','YYYY-MM-DD'))
,
partition item116 values less than
(to_date('1997-01-01','YYYY-MM-DD'))
,
partition item117 values less than
(to_date('1997-01-16','YYYY-MM-DD'))
,
partition item118 values less than
(to_date('1997-02-01','YYYY-MM-DD'))
,
partition item119 values less than
(to_date('1997-02-16','YYYY-MM-DD'))
,
partition item120 values less than
(to_date('1997-03-01','YYYY-MM-DD'))
,
partition item121 values less than
(to_date('1997-03-16','YYYY-MM-DD'))
,
partition item122 values less than
(to_date('1997-04-01','YYYY-MM-DD'))
,
partition item123 values less than
(to_date('1997-04-16','YYYY-MM-DD'))
,
partition item124 values less than
(to_date('1997-05-01','YYYY-MM-DD'))
,
partition item125 values less than
(to_date('1997-05-16','YYYY-MM-DD'))
,
partition item126 values less than
(to_date('1997-06-01','YYYY-MM-DD'))
,
partition item127 values less than
(to_date('1997-06-16','YYYY-MM-DD'))
,
partition item128 values less than
(to_date('1997-07-01','YYYY-MM-DD'))
,
partition item129 values less than
(to_date('1997-07-16','YYYY-MM-DD'))
,
partition item130 values less than
(to_date('1997-08-01','YYYY-MM-DD'))
,
partition item131 values less than
(to_date('1997-08-16','YYYY-MM-DD'))
,
partition item132 values less than
(to_date('1997-09-01','YYYY-MM-DD'))
,
partition item133 values less than
(to_date('1997-09-16','YYYY-MM-DD'))
,
partition item134 values less than
(to_date('1997-10-01','YYYY-MM-DD'))
,
partition item135 values less than
(to_date('1997-10-16','YYYY-MM-DD'))
,
partition item136 values less than
(to_date('1997-11-01','YYYY-MM-DD'))
,
partition item137 values less than
(to_date('1997-11-16','YYYY-MM-DD'))
,
partition item138 values less than
(to_date('1997-12-01','YYYY-MM-DD'))
,
partition item139 values less than
(to_date('1997-12-16','YYYY-MM-DD'))
,
partition item140 values less than
(to_date('1998-01-01','YYYY-MM-DD'))
,
partition item141 values less than
(to_date('1998-01-16','YYYY-MM-DD'))
,
partition item142 values less than
(to_date('1998-02-01','YYYY-MM-DD'))
,
partition item143 values less than
(to_date('1998-02-16','YYYY-MM-DD'))
,
partition item144 values less than
(to_date('1998-03-01','YYYY-MM-DD'))
,
partition item145 values less than
(to_date('1998-03-16','YYYY-MM-DD'))
,
partition item146 values less than
(to_date('1998-04-01','YYYY-MM-DD'))
,
partition item147 values less than
(to_date('1998-04-16','YYYY-MM-DD'))
,
partition item148 values less than
(to_date('1998-05-01','YYYY-MM-DD'))
,
partition item149 values less than
(to_date('1998-05-16','YYYY-MM-DD'))
,
partition item150 values less than
(to_date('1998-06-01','YYYY-MM-DD'))
,
partition item151 values less than
(to_date('1998-06-16','YYYY-MM-DD'))
```

```
,
partition item152 values less than
(to_date('1998-07-01','YYYY-MM-DD'))
,
partition item153 values less than
(to_date('1998-07-16','YYYY-MM-DD'))
,
partition item154 values less than
(to_date('1998-08-01','YYYY-MM-DD'))
,
partition item155 values less than
(to_date('1998-08-16','YYYY-MM-DD'))
,
partition item156 values less than
(to_date('1998-09-01','YYYY-MM-DD'))
,
partition item157 values less than
(to_date('1998-09-16','YYYY-MM-DD'))
,
partition item158 values less than
(to_date('1998-10-01','YYYY-MM-DD'))
,
partition item159 values less than
(to_date('1998-11-01','YYYY-MM-DD'))
,
partition item160 values less than (MAXVALUE)
)
with rowid
(
    c_nationkey           ,
    c_custkey            ,
    l_returnflag         ,
    s_nationkey         ,
    o_orderdate          ,
    o_ajm                 ,
    c_ajm                ,
    s_ajm                ,
    l_extendedprice       ,
    l_discount           ,
    l_shipdate
)
including new values
;
}
*sql
{
connect tpcd/tpcd;
drop materialized view ma_locs;
create materialized view ma_locs
pctfree 2
tablespace ts_ma_lineitem
storage (initial 512k next 512k maxextents
unlimited pctincrease 0)
parallel (degree 1)
nologging
BUILD immediate
REFRESH fast
ON commit
enable query rewrite
as select
  dfy(l_shipdate) as ship_dfy,
  dfy(o_orderdate) as order_dfy,
  c_nationkey,
  s_nationkey,
  o_ajm,
  c_ajm,
  s_ajm,
  sum(l_extendedprice * (1 - l_discount)) as
volume,
  count(l_extendedprice * (1 - l_discount)) as
count_volume,
  count(*) as count_grp
FROM
    mj_locs
GROUP BY
  dfy(l_shipdate),
  dfy(o_orderdate),
  c_nationkey,
  s_nationkey,
  o_ajm,
  c_ajm,
  s_ajm
;
}
*sql
{
connect tpcd/tpcd;
drop materialized view mj_ppssrn;
create materialized view mj_ppssrn
pctfree 0
tablespace ts_mj_ppssrn
storage (initial 32k next 4m pctincrease 0)
parallel (degree 20)
nologging
partition by range (r_name)
subpartition by hash(p_size)
subpartitions 64
(
partition mj_ppssrn1 values less than ('AMERICA')
,
partition mj_ppssrn2 values less than ('ASIA')
,
partition mj_ppssrn3 values less than ('EUROPE')
,
partition mj_ppssrn4 values less than ('MIDDLE
EAST')
,
partition mj_ppssrn5 values less than (MAXVALUE)
)
enable row movement
BUILD immediate
REFRESH fast
ON commit
enable query rewrite
as select
/*+ user_hash(ps, s, n, r, p) ordered */
  n_name,
  ajm(n.rowid) as n_ajm,
  p_brand,
  p_mfgr,
  p_name,
  p_size,
  p_type,
  p_partkey,
  ajm(p.rowid) as p_ajm,
  ps_availqty,
  ps_partkey,
  ps_suppkey,
  ps_supplycost,
  s_suppkey,
  ajm(s.rowid) as s_ajm,
  ajm(r.rowid) as r_ajm,
  r_name ,
  p.rowid as p_rowid,
  ps.rowid as ps_rowid,
  s.rowid as s_rowid,
  n.rowid as n_rowid,
  r.rowid as r_rowid
```

```
FROM
  parts p,
  supplier s,
  partsupp ps,
  nation n,
  region r
WHERE
  ps_partkey = p_partkey(+)
  AND   ps_suppkey = s_suppkey(+)
  AND   s_nationkey = n_nationkey(+)
  AND   n_regionkey = r_regionkey(+)
;
}
*wait
*wait
*bgoff
%e-scuvo
```

100g_816_sdgen.dat

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

%b-preproc

*sql
\svrmgrl <<!
\set echo on;
\set termout on;
\spool phase#.lst;
\connect internal/oracle;
\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;
\!

*load
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-sdgen
*bgon=30
################################################
############################
# Data (Seed File) Generation Phase
*time=Begin creating seed files
*dbgen
{
-v -O s -s 100 -C 5
}
*time=Done creating seed files for degree 5
*wait
*time=Done creating seed files
*wait
*wait
*bgoff
%e-sdgen
```

change_degree30.sql

```
spool change_degree.out
set echo on;
connect tpcd/tpcd;

alter table LINEITEM storage (next 1m);
alter table ORDERS   storage (next 1m);
alter table MJ_LOCS  storage (next 1m);

alter table ORDERS   parallel (degree 30);
alter table CUSTOMER parallel (degree 30);
alter table LINEITEM parallel (degree 30);
alter table NATION   NOPARALLEL;
alter table REGION   NOPARALLEL;
alter table ORDERS   parallel (degree 30);
alter table PARTS    NOPARALLEL;
alter table PARTSUPP parallel (degree 30);
alter table SUPPLIER parallel (degree 30);

alter table MA_L_SDFY       NOPARALLEL;
alter table MA_LINEITEM  NOPARALLEL;
alter table MA_LOCS         NOPARALLEL;
alter table MJ_LOCS        parallel (degree 30);
alter table MJ_PPSSRN      parallel (degree 30);

alter table MLOG$_LINEITEM parallel (degree 30);
alter table MLOG$_ORDERS   parallel (degree 30);
alter table MLOG$_MJ_LOCS  parallel (degree 30);

alter table MLOG$_CUSTOMER parallel (degree 30);
alter table MLOG$_NATION   parallel (degree 30);
alter table MLOG$_PARTS    parallel (degree 30);
alter table MLOG$_PARTSUPP parallel (degree 30);
alter table MLOG$_REGION   parallel (degree 30);
alter table MLOG$_SUPPLIER parallel (degree 30);

alter index IJ_L_NULLRID        parallel (degree
30);
alter index IJ_L_ORID           parallel (degree
30);
alter index IJ_L_LRID           parallel (degree
30);
alter index IJ_L_PKEY           parallel (degree
30);
alter index IJ_L_RDATE          parallel (degree
30);
alter index IJ_L_SDATE          parallel (degree
30);
alter index IJ_L_ODATE          parallel (degree
30);
alter index I_P_PARTKEY         parallel (degree
30);
alter index I_PS_PARTKEY_SUPPKEY NOPARALLEL ;
alter index I_C_CUSTKEY         parallel (degree
30);
alter index I_S_SUPPKEY         parallel (degree
30);
alter index I_R_REGIONKEY        NOPARALLEL ;
alter index I_N_NATIONKEY        NOPARALLEL ;
alter index P_CBP               parallel (degree
30);
alter index P_TP                parallel (degree
30);
alter index O_OKEY              parallel (degree
30);
alter index O_CLOKOD            parallel (degree
```

```
8);
alter index L_ORED              parallel (degree
30);
alter index IJ_P_NNPSPS         parallel (degree
30);
alter index IJ_P_PPSN           parallel (degree
30);

spool off;
```

create_all.bat

```
REM
REM  create_all.bat
REM

rem echo ------ start plcre phase
rem perl bumpx.pl -s -x -o 100g_816_plcre.dat -p
plcre
rem echo ------ end plcre phase
rem mv *.lst load-log/plcre

rem echo ------ start dbcre phase
rem perl bumpx.pl -s -x -o 100g_816_dbcre.dat -p
dbcre
rem echo ------ end dbcre phase
rem mv *.lst load-log/dbcre

rem echo ------ start sctso phase
rem perl bumpx.pl -s -x -o 100g_816_sctso.dat -p
sccre
rem echo ------ end sctso phase
rem mv *.lst load-log/sctso

date/t >> create_all.out
time/t >> create_all.out
echo ---------- start creating database ---------
- >> create_all.out
rem echo ---------- re-start creating database --
-------- >> create_all.out
echo   >> create_all.out


date/t >> create_all.out
time/t >> create_all.out
echo ------ start scuto phase >> create_all.out
perl bumpx.pl -s -x -o 100g_816_scuto.dat -p
sccre
date/t >> create_all.out
time/t >> create_all.out
echo ------ end scuto phase >> create_all.out
mv *.lst load-log/scuto

date/t >> create_all.out
time/t >> create_all.out
echo ------ start dapop phase >> create_all.out
perl bumpx.pl -s -x -o 100g_816_dapop.dat -p
dapop
date/t >> create_all.out
time/t >> create_all.out
echo ------ end dapop phase >> create_all.out
mv *.lst load-log/dapop

date/t >> create_all.out
time/t >> create_all.out
echo ------ start scuvo phase >> create_all.out
perl bumpx.pl -s -x -o 100g_816_scuvo.dat -p
scuvo
date/t >> create_all.out
time/t >> create_all.out
echo ------ end scuvo phase >> create_all.out
mv *.lst load-log/scuvo

date/t >> create_all.out
time/t >> create_all.out
echo ------ start anlyz phase >> create_all.out
perl bumpx.pl -s -x -o 100g_816_anlyz.dat -p
anlyz
date/t >> create_all.out
time/t >> create_all.out
echo ------ end anlyz phase >> create_all.out
mv *.lst load-log/anlyz

date/t >> create_all.out
time/t >> create_all.out
echo ------ start ixcre phase >> create_all.out
perl bumpx.pl -s -x -o 100g_816_ixcre.dat -p
ixcre
date/t >> create_all.out
time/t >> create_all.out
echo ------ end ixcre phase >> create_all.out
mv *.lst load-log/ixcre

date/t >> create_all.out
time/t >> create_all.out
echo ------ start change degree >> create_all.out
svrmgrl @change_degree30.sql
date/t >> create_all.out
time/t >> create_all.out
echo ------ end change degree >> create_all.out

date/t >> create_all.out
time/t >> create_all.out
echo ------ end of the AUDIT 100GB DB LOAD phase
>> create_all.out
```

links100g.prn

```
log_100g1         \device\harddisk6\partition1
log_100g2         \device\harddisk6\partition2
sys_100g1         \device\harddisk3\partition3
sys_100g2         \device\harddisk4\partition1
sys_100g3         \device\harddisk9\partition1
sys_100g4         \device\harddisk10\partition1
sys_100g5         \device\harddisk11\partition1
ts_11    \device\harddisk3\partition21
ts_12    \device\harddisk4\partition19
ts_13    \device\harddisk9\partition19
ts_14    \device\harddisk10\partition19
ts_15    \device\harddisk11\partition19
ts_c_100g1        \device\harddisk3\partition9
ts_c_100g2        \device\harddisk4\partition7
ts_c_100g3        \device\harddisk9\partition7
ts_c_100g4        \device\harddisk10\partition7
ts_c_100g5        \device\harddisk11\partition7
ts_constr_100g1   \device\harddisk3\partition32
ts_constr_100g2   \device\harddisk4\partition30
ts_constr_100g3   \device\harddisk9\partition30
ts_constr_100g4   \device\harddisk10\partition30
ts_constr_100g5   \device\harddisk11\partition30
ts_default_100g1  \device\harddisk3\partition7
ts_default_100g2  \device\harddisk4\partition5
ts_default_100g3  \device\harddisk9\partition5
ts_default_100g4  \device\harddisk10\partition5
ts_default_100g5  \device\harddisk11\partition5
ts_ij_l_odate_100g1
         \device\harddisk3\partition24
```

```
ts_ij_l_odate_100g2
        \device\harddisk4\partition22
ts_ij_l_odate_100g3
        \device\harddisk9\partition22
ts_ij_l_odate_100g4
        \device\harddisk10\partition22
ts_ij_l_odate_100g5
        \device\harddisk11\partition22
ts_ij_l_pkey_100g1
        \device\harddisk3\partition25
ts_ij_l_pkey_100g2
        \device\harddisk4\partition23
ts_ij_l_pkey_100g3
        \device\harddisk9\partition23
ts_ij_l_pkey_100g4
        \device\harddisk10\partition23
ts_ij_l_pkey_100g5
        \device\harddisk11\partition23
ts_ij_l_rdate_100g1
        \device\harddisk3\partition23
ts_ij_l_rdate_100g2
        \device\harddisk4\partition21
ts_ij_l_rdate_100g3
        \device\harddisk9\partition21
ts_ij_l_rdate_100g4
        \device\harddisk10\partition21
ts_ij_l_rdate_100g5
        \device\harddisk11\partition21
ts_ij_l_sdate_100g1
        \device\harddisk3\partition22
ts_ij_l_sdate_100g2
        \device\harddisk4\partition20
ts_ij_l_sdate_100g3
        \device\harddisk9\partition20
ts_ij_l_sdate_100g4
        \device\harddisk10\partition20
ts_ij_l_sdate_100g5
        \device\harddisk11\partition20
ts_ij_lrid_100g1  \device\harddisk3\partition28
ts_ij_lrid_100g2  \device\harddisk4\partition26
ts_ij_lrid_100g3  \device\harddisk9\partition26
ts_ij_lrid_100g4  \device\harddisk10\partition26
ts_ij_lrid_100g5  \device\harddisk11\partition26
ts_ij_nullrid_100g1
        \device\harddisk3\partition30
ts_ij_nullrid_100g2
        \device\harddisk4\partition28
ts_ij_nullrid_100g3
        \device\harddisk9\partition28
ts_ij_nullrid_100g4
        \device\harddisk10\partition28
ts_ij_nullrid_100g5
        \device\harddisk11\partition28
ts_ij_orid_100g1  \device\harddisk3\partition29
ts_ij_orid_100g2  \device\harddisk4\partition27
ts_ij_orid_100g3  \device\harddisk9\partition27
ts_ij_orid_100g4  \device\harddisk10\partition27
ts_ij_orid_100g5  \device\harddisk11\partition27
ts_ij_p_nnpsps_100g1
        \device\harddisk3\partition34
ts_ij_p_nnpsps_100g2
        \device\harddisk4\partition32
ts_ij_p_nnpsps_100g3
        \device\harddisk9\partition32
ts_ij_p_nnpsps_100g4
        \device\harddisk10\partition32
ts_ij_p_nnpsps_100g5
        \device\harddisk11\partition32
ts_ij_p_ppsn_100g1
        \device\harddisk3\partition33
ts_ij_p_ppsn_100g2
        \device\harddisk4\partition31
ts_ij_p_ppsn_100g3
        \device\harddisk9\partition31
ts_ij_p_ppsn_100g4
        \device\harddisk10\partition31
ts_ij_p_ppsn_100g5
        \device\harddisk11\partition31
ts_ind_lored_100g1
        \device\harddisk3\partition26
ts_ind_lored_100g2
        \device\harddisk4\partition24
ts_ind_lored_100g3
        \device\harddisk9\partition24
ts_ind_lored_100g4
        \device\harddisk10\partition24
ts_ind_lored_100g5
        \device\harddisk11\partition24
ts_ind_oclokod_100g1
        \device\harddisk3\partition27
ts_ind_oclokod_100g2
        \device\harddisk4\partition25
ts_ind_oclokod_100g3
        \device\harddisk9\partition25
ts_ind_oclokod_100g4
        \device\harddisk10\partition25
ts_ind_oclokod_100g5
        \device\harddisk11\partition25
ts_ind_okey_100g1 \device\harddisk3\partition31
ts_ind_okey_100g2 \device\harddisk4\partition29
ts_ind_okey_100g3 \device\harddisk9\partition29
ts_ind_okey_100g4 \device\harddisk10\partition29
ts_ind_okey_100g5 \device\harddisk11\partition29
ts_l_100g1        \device\harddisk3\partition8
ts_l_100g2        \device\harddisk4\partition6
ts_l_100g3        \device\harddisk9\partition6
ts_l_100g4        \device\harddisk10\partition6
ts_l_100g5        \device\harddisk11\partition6
ts_ma_lineitem_100g1
        \device\harddisk3\partition14
ts_ma_lineitem_100g2
        \device\harddisk4\partition12
ts_ma_lineitem_100g3
        \device\harddisk9\partition12
ts_ma_lineitem_100g4
        \device\harddisk10\partition12
ts_ma_lineitem_100g5
        \device\harddisk11\partition12
ts_maint_100g1    \device\harddisk3\partition19
ts_maint_100g2    \device\harddisk4\partition17
ts_maint_100g3    \device\harddisk9\partition17
ts_maint_100g4    \device\harddisk10\partition17
ts_maint_100g5    \device\harddisk11\partition17
ts_mj_locs_100g1  \device\harddisk3\partition15
ts_mj_locs_100g10 \device\harddisk11\partition14
ts_mj_locs_100g2  \device\harddisk4\partition13
ts_mj_locs_100g3  \device\harddisk9\partition13
ts_mj_locs_100g4  \device\harddisk10\partition13
ts_mj_locs_100g5  \device\harddisk11\partition13
ts_mj_locs_100g6  \device\harddisk3\partition16
ts_mj_locs_100g7  \device\harddisk4\partition14
ts_mj_locs_100g8  \device\harddisk9\partition14
ts_mj_locs_100g9  \device\harddisk10\partition14
ts_mj_ppssrn_100g1
        \device\harddisk3\partition20
ts_mj_ppssrn_100g2
        \device\harddisk4\partition18
```

```
ts_mj_ppssrn_100g3
        \device\harddisk9\partition18
ts_mj_ppssrn_100g4
        \device\harddisk10\partition18
ts_mj_ppssrn_100g5
        \device\harddisk11\partition18
ts_mllog_100g1    \device\harddisk3\partition17
ts_mllog_100g2    \device\harddisk4\partition15
ts_mllog_100g3    \device\harddisk9\partition15
ts_mllog_100g4    \device\harddisk10\partition15
ts_mllog_100g5    \device\harddisk11\partition15
ts_molog_100g1    \device\harddisk3\partition18
ts_molog_100g2    \device\harddisk4\partition16
ts_molog_100g3    \device\harddisk9\partition16
ts_molog_100g4    \device\harddisk10\partition16
ts_molog_100g5    \device\harddisk11\partition16
ts_o_100g1        \device\harddisk3\partition10
ts_o_100g2        \device\harddisk4\partition8
ts_o_100g3        \device\harddisk9\partition8
ts_o_100g4        \device\harddisk10\partition8
ts_o_100g5        \device\harddisk11\partition8
ts_p_100g1        \device\harddisk3\partition11
ts_p_100g2        \device\harddisk4\partition9
ts_p_100g3        \device\harddisk9\partition9
ts_p_100g4        \device\harddisk10\partition9
ts_p_100g5        \device\harddisk11\partition9
ts_p_cbp_100g1    \device\harddisk3\partition36
ts_p_cbp_100g2    \device\harddisk4\partition34
ts_p_cbp_100g3    \device\harddisk9\partition34
ts_p_cbp_100g4    \device\harddisk10\partition34
ts_p_cbp_100g5    \device\harddisk11\partition34
ts_p_tp_100g1     \device\harddisk3\partition35
ts_p_tp_100g2     \device\harddisk4\partition33
ts_p_tp_100g3     \device\harddisk9\partition33
ts_p_tp_100g4     \device\harddisk10\partition33
ts_p_tp_100g5     \device\harddisk11\partition33
ts_ps_100g1       \device\harddisk3\partition12
ts_ps_100g2       \device\harddisk4\partition10
ts_ps_100g3       \device\harddisk9\partition10
ts_ps_100g4       \device\harddisk10\partition10
ts_ps_100g5       \device\harddisk11\partition10
ts_s_100g1        \device\harddisk3\partition13
ts_s_100g2        \device\harddisk4\partition11
ts_s_100g3        \device\harddisk9\partition11
ts_s_100g4        \device\harddisk10\partition11
ts_s_100g5        \device\harddisk11\partition11
ts_temp_100g1     \device\harddisk3\partition5
ts_temp_100g2     \device\harddisk4\partition3
ts_temp_100g3     \device\harddisk9\partition3
ts_temp_100g4     \device\harddisk10\partition3
ts_temp_100g5     \device\harddisk11\partition3
ts_temp_100g6     \device\harddisk3\partition6
ts_temp_100g7     \device\harddisk4\partition4
ts_temp_100g8     \device\harddisk9\partition4
ts_temp_100g9     \device\harddisk10\partition4
ts_temp_100g10    \device\harddisk11\partition4
undo_100g1        \device\harddisk3\partition4
undo_100g2        \device\harddisk4\partition2
undo_100g3        \device\harddisk9\partition2
undo_100g4        \device\harddisk10\partition2
undo_100g5        \device\harddisk11\partition2
ts_temp_100g11  \device\harddisk4\partition35
ts_temp_100g12    \device\harddisk3\partition37
ts_temp_100g13    \device\harddisk9\partition35
ts_temp_100g14    \device\harddisk10\partition35
ts_temp_100g15    \device\harddisk11\partition35
```

driver.c

```
/* Sccsid:     @(#)driver.c     9.1.1.34
5/1/96  11:45:07 */
/* 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, long start, long count,
long upd_num);
```

```
int pr_drange (int tbl, long min, long cnt, long
num);
int set_files (int t, int pload);
void seed_name (char *tgt, long s, long c, long
p);
int partial (int, int);
void gen_seeds (int start, int s);

#ifndef ORIG
#include "part.h"
int partition = 0;      /* partition flag
*/
vals *values;
FILE *in;               /* fdes for input file
*/
int  *out;              /* output fd   array
*/
FILE *par;              /* fdes for params file
*/
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 /* !ORIG */

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       child;           -- non-zero if
there is an associated
 detail table
 * }         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_time (dss_time_t * t, 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_time (dss_time_t * t, 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 (long skip_count);
long sd_line (long skip_count);
long sd_order (long skip_count);
long sd_part (long skip_count);
long sd_psupp (long skip_count);
long sd_supp (long skip_count);
long sd_order_line (long skip_count);
long sd_part_psupp (long skip_count);
long sd_nation (long skip_count);
long sd_region (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_time (FILE * f);
int hd_nation (FILE * f);
int hd_region (FILE * f);
```

```
tdef tdefs[] =
{
  {"part.tbl", "part table", 200000, hd_part,
   {pr_part, ld_part}, sd_part, NONE},
  {"partsupp.tbl", "partsupplier table", 200000,
hd_psupp,
   {pr_psupp, ld_psupp}, sd_psupp, NONE},
  {"supplier.tbl", "suppliers table", 10000,
hd_supp,
   {pr_supp, ld_supp}, sd_supp, NONE},
  {"customer.tbl", "customers table", 150000,
hd_cust,
   {pr_cust, ld_cust}, sd_cust, NONE},
  {"order.tbl", "order table", 150000, hd_order,
   {pr_order, ld_order}, sd_order, NONE},
  {"lineitem.tbl", "lineitem table", 150000,
hd_line,
   {pr_line, ld_line}, sd_line, NONE},
  {"order.tbl", "order/lineitem tables", 150000,
hd_order_line,
   {pr_order_line, ld_order_line}, sd_order_line,
LINE},
  {"part.tbl", "part/partsupplier tables",
200000, hd_part_psupp,
   {pr_part_psupp, ld_part_psupp}, sd_part_psupp,
PSUPP},
  {"time.tbl", "time table", 2557, hd_time,
   {pr_time, ld_time}, NO_LFUNC, NONE},
  {"nation.tbl", "nation table", NATIONS_MAX,
hd_nation,
   {pr_nation, ld_nation}, sd_nation, NONE},
  {"region.tbl", "region table", NATIONS_MAX,
hd_region,
   {pr_region, ld_region}, sd_region, NONE},
  {"TPCDSEED", NULL, 0, NO_FUNC,
   {NO_FUNC, NO_FUNC}, NO_LFUNC, NONE}
};

int *pids;
#define LIFENOISE(n)       if (verbose && (i % n)
== 0) fprintf(stderr, ".")

void
mk_sparse (long res[], long base[], long seq)
{
  long low_mask, seq_mask, overflow = 0;
  int count = 0;

  low_mask = (1 << SPARSE_KEEP) - 1;
  seq_mask = (1 << SPARSE_BITS) - 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[0] & low_mask, res);
  bcd2_bin (&low_mask, res[0]);
  bcd2_bin (&seq_mask, res[1]);
  res[0] = low_mask;
  res[1] = seq_mask;
  return;
}

/*
 * 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 + 1);
           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);

}

/*
 * generate a particular table
 */
#ifndef SUPPORT_64BITS
void
gen_h_tbl (int tnum, long start[], long count[],
long upd_num)
{
  order_t o;
  static long max_easy = LONG_MAX >> SPARSE_BITS;
  HUGE_T (i);
  HUGE_T (sk);

  HUGE_SET (start, i);
  while (HUGE_CMP (count, 0) > 0)
        {
          if (verbose && (i[0] % 1000) == 0)
                fprintf (stderr, ".");
          mk_sparse (sk, i,
                                     (upd_num ==
0) ? 0 : 1 + upd_num / (10000 / refresh));
          mk_order (sk, &o);
          tdefs[tnum].loader[direct] (&o,
upd_num);
          HUGE_SUB (count, 1, count);
          HUGE_ADD (i, 1, i);
        }
  return;
}
#endif /* SUPPORT_64BITS */

#ifndef SUPPORT_64BITS
void
gen_tbl (int tnum, DSS_HUGE start, DSS_HUGE
count, long upd_num)
#else
void
gen_tbl (int tnum, long start, long count, long
upd_num)
#endif
                  /* SUPPORT_64BITS */
{
  order_t o;
  supplier_t supp;
  customer_t cust;
  part_t part;
  dss_time_t t;
  code_t code;
  static long max_easy = LONG_MAX >> SPARSE_BITS;
  static int completed = 0;
  long i;
  HUGE_T (sk);
  HUGE_T (temp);

  for (i = start; count; count--, i++)
        {
          LIFENOISE (1000);
          ROW_START (tnum);
          switch (tnum)
                {
                case LINE:
                case ORDER:
                case ORDER_LINE:
                  temp[0] = i;
                  temp[1] = 0;
                 if (i > max_easy)
                         mk_sparse (sk, temp,

(upd_num == 0) ? 0 : 1 + upd_num / (10000 /
refresh));
                  else
                  {
                  sk[1] = MK_SPARSE(i,
                      (upd_num == 0)?0:1 +
upd_num/(10000 / refresh));
                  sk[0] = 0;
                  }
                  mk_order (sk, &o);
                  tdefs[tnum].loader[direct]
(&o, upd_num);
                  break;
                case SUPP:
                  mk_supp (i, &supp);
                  tdefs[tnum].loader[direct]
(&supp, upd_num);
                  break;
                case CUST:
                  mk_cust (i, &cust);
                  tdefs[tnum].loader[direct]
(&cust, upd_num);
                  break;
                case PSUPP:
                case PART:
                case PART_PSUPP:
                  mk_part (i, &part);
                  tdefs[tnum].loader[direct]
(&part, upd_num);
                  break;
                case TIME:
                  mk_time (i, &t);
                  tdefs[tnum].loader[direct]
(&t, 0);
                  break;
                case NATION:
                  mk_nation (i, &code);
```

```
                    tdefs[tnum].loader[direct]
(&code, 0);
                    break;
                  case REGION:
                    mk_region (i, &code);
                    tdefs[tnum].loader[direct]
(&code, 0);
                    break;
                  }
            ROW_STOP (tnum);
          }
  completed |= 1 << tnum;
}



void
usage (void)
{
  fprintf (stderr, "%s\n%s\n\t%s\n%s %s\n\n",
                    "USAGE:",
                    "dbgen [-{vfFD}] [-O
{fhmst}][-T {pcsoPSOL}]",
                    "[-s <scale>][-C <procs>][-S
<step>]",
                    "dbgen [-v] [-O {dhmrt}] [-s
<scale>]",
                    "[-U <updates>] [-r
<percent>]");
  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, "-f     -- force. Overwrite
existing files\n");
  fprintf (stderr, "-F     -- generate flat files
output\n");
  fprintf (stderr, "-h     -- display this
message\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 s   -- generate seed sets
ONLY\n");
  fprintf (stderr, "-O t   -- use TIME table and
julian dates\n");
  fprintf (stderr, "-r <n> -- updates refresh
(n/100)%% of the\n");
  fprintf (stderr, "          data set\n");
  fprintf (stderr, "-R <n> -- resume seed rfile
generation with step <n>\n");
  fprintf (stderr, "-s <n> -- set Scale Factor
(SF) to  <n> \n");
  fprintf (stderr, "-S <n> -- build the <n>th
step of the data 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");
#ifndef ORIG
    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, "<col1 type>\n");
    fprintf(stderr, "<col1 pos> - not used at
this point\n");
    fprintf(stderr, "<col1 val1>\n");
    fprintf(stderr, "<col1 val2>\n");
    fprintf(stderr, "...\n");
    fprintf(stderr, "<col1 valm>\n");
    fprintf(stderr, "Notes:\n");
    fprintf(stderr, "%s\n",
             "Column types are: 0 for numbers, 1
for strings, 2 for dates");
    fprintf(stderr, "The direct load option is
automatically turned on.\n");
    fprintf(stderr,
"=================================\n\n");
#endif /* !ORIG */
  fprintf (stderr,
                    "\nTo generate the SF=1
(1GB) database population , use:\n");
  fprintf (stderr, "\tdbgen -vfF -s 1\n");
  fprintf (stderr, "\n%s %s\n",
                    "To generate the
qualification database population",
                    "(100 MB), use:\n");
  fprintf (stderr, "\tdbgen -vfF -s 0.1\n");
  fprintf (stderr, "\nTo generate updates for a
SF=1 (1GB), use:\n");
  fprintf (stderr, "\tdbgen -v -O s -s 1\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)
{
  HUGE_T (h_rowcnt);
  HUGE_T (h_minrow);
  long rowcnt;
  long minrow;
  char fname[80];
  long extrarows;


  if (verbose)
        {
          fprintf (stderr, "Starting to load
stage %d of %d of %s...",
                                s + 1, children,
tdefs[tbl].comment);
        }
  if (load_state (scale, children, s))
        {
          seed_name (fname, scale, children, s);
          fprintf (stderr, "Unable to load seeds
(%s)\n", fname);
          exit (-1);
        }

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

#ifndef SUPPORT_64BITS
  if ((tbl == LINE || tbl == ORDER) && scale >
MAX_32B_SCALE)
        {
          long step_size;

          step_size = scale / children;
          if (step_size > MAX_32B_SCALE)
                {
                  fprintf (stderr, "Each child
must generate less than 1TB.\n");
                  fprintf (stderr,
                                        "Please
rerun DBGEN with a larger number of children\n");
                  exit (1);
                }
          rowcnt = step_size * tdefs[tbl].base;
          extrarows = tdefs[tbl].base %
children;
          extrarows *= scale;
          extrarows %= children;
          LONG2HUGE (rowcnt, h_rowcnt);
          LONG2HUGE (rowcnt, h_minrow);
          HUGE_MUL (h_minrow, s);
          if (s == children)
                HUGE_ADD (h_rowcnt, extrarows,
h_rowcnt);
          gen_h_tbl (tbl, h_minrow, h_rowcnt,
upd_num);
        }
 else
        {
#endif /* SUPPORT_64BITS */
          rowcnt = tdefs[tbl].base * scale;
          extrarows = rowcnt % children;
          rowcnt /= children;
          minrow = rowcnt * s + 1;
          if (s == children)
                rowcnt += extrarows;
          gen_tbl (tbl, minrow, rowcnt,
upd_num);
#ifndef SUPPORT_64BITS
        }
#endif /* SUPPORT_64BITS */

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

  return (0);
}


int
pload (int tbl)
{
  int c = 0, i, status;
  char cmdline[256];

  rowcnt = tdefs[tbl].base * scale;
  if (rowcnt % children)
        {
          fprintf (stderr, "'-C' cannot split
load equally\n");
          exit (-1);
        }
  else
        rowcnt /= children;
  if (verbose)
        {
          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);
                  exit (0);
                }
          else if (verbose)
        /* PARENT */
                  fprintf (stderr, ".");
        }

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

  c = children;
```

```
  while (c)
        {
          i = WAIT (&status, pids[c - 1]);
          if (i == -1 && children)
                {
                  if (errno == ECHILD)
                        fprintf (stderr, "Could
not wait on pid %d\n", pids[c - 1]);
                  else if (errno == EINTR)
                        fprintf (stderr,
"Process %d stopped abnormally\n", pids[c - 1]);
                  else if (errno == EINVAL)
                        fprintf (stderr,
"Program bug\n");
                }
          if (status & 0xFF)
                {
                  if (status & 0xFF == 0117)
                        printf ("Process %d:
STOPPED\n", i);
                  else
                        printf ("Process %d:
rcvd signal %d\n",
                                                i,
status & 0x7F);
                }
          c--;
        }

  if (direct == 0)
        {
          sprintf (cmdline, "mv %s/pl%d.%d%c
%s/%s",
                                env_config
(PATH_TAG, PATH_DFLT), pids[0], tbl,
                                'B' +
tdefs[tbl].child,
                                env_config
(PATH_TAG, PATH_DFLT), tdefs[tbl].name);
          system (cmdline);
#if (defined(WIN32)&&!defined(_POSIX_))
          sprintf (cmdline, "COPY %s\\pl*.%d
%s\\%s",
                                env_config
(PATH_TAG, PATH_DFLT), tbl,
                                env_config
(PATH_TAG, PATH_DFLT), tdefs[tbl].name);

          fprintf (stderr, "%s\n", cmdline);
          system (cmdline);

          sprintf (cmdline, "DEL %s\\pl*.%d",
                                env_config
(PATH_TAG, PATH_DFLT), tbl);
          fprintf (stderr, "%s\n", cmdline);
          system (cmdline);

#else
          for (c = 1; c < children; c++)
                {
                  sprintf (cmdline,
                                        "cat
%s/pl%d.%d%c >> %s/%s; rm %s/pl%d.%d%c",
                                        env_config
(PATH_TAG, PATH_DFLT), pids[c], tbl,
                                        'B' +
tdefs[tbl].child,
                                        env_config
(PATH_TAG, PATH_DFLT), tdefs[tbl].name,
                                                env_config
(PATH_TAG, PATH_DFLT), pids[c], tbl,
                                                'B' +
tdefs[tbl].child);
                        system (cmdline);
                }
#endif /* WIN32 */
        }

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

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

#ifndef ORIG
    /* set some default */
    if (partition) {
      (void) sprintf(param,"%s","./part.param");
      (void) sprintf(outname,"%s","/dev/sqlldr");
    }
#endif /* !ORIG */

#ifndef DOS
#ifndef ORIG
    while ((option = getopt(count, vector,
"C:DFfhn:O:P:r:R:s:S:T:U:vi:o:a:p")) != -1)
#else
    while ((option = getopt(count, vector,
"C:DFfhn:O:P:r:R:s:S:T:U:v")) != -1)
#endif /* !ORIG */
#else
  while ((option = getopt (count, vector,
"DFfhn:O:P:r:R:s:S:T:U:v")) != -1)
#endif /* !DOS */

         switch (option)
           {
#ifndef ORIG
             case 'p':
               partition = 1;
               direct = 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 /* !ORIG */
           case 'S':
         /* generate a particular STEP */
                  step = atoi (optarg) - 1;
                  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;
                  case 't':
        /* generate time ONLY */
                        table = 1 << TIME;
                        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 <
TIME; i++)
                          {
                                tdefs[i].base
*= flt_scale;
                                if
(tdefs[i].base < 1)

tdefs[i].base = 1;
                          }
                  }
                else
                  scale = (long) flt_scale;
                if (scale > MAX_SCALE &&
bld_seeds != 1)
                  {
                        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':
        /* generate seed sets */
                        bld_seeds = 1;
                        break;
                    case 't':
        /* use TIME table and join fields */
                        oldtime = 1;
                        if (updates == 0)
                          table |= (1 << TIME);
                        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;
            case 'R':
        /* resume seed file generation */
                  resume = atoi (optarg);
                  break;
#ifndef DOS
            case 'C':
                  children = atoi (optarg);
                  if (children > 999)
        /* limitation of current seed file names
*/
                    {
                        printf ("Child process
counts of > 999 not supported.\n");
                        exit (1);
                    }
                  pids = malloc (children *
sizeof (pid_t));
                  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,
                                        "TPC-D
Population Generator (Version %d.%d.%d%s)\n",
                                        VERSION,
RELEASE,
                                        MODIFICATION,
PATCH);
                  fprintf (stderr, "Copyright %s
%s\n", TPC, C_DATES);
                  usage ();
                  exit (1);
              }
    return;
}

void
gen_seeds (int start, int s)
{
int i;
long c;
double step_size;

if (start != 1)
          {
          if (load_state (s, children, start - 1))
                  {
                  char fname[80];
                  seed_name (fname, s, children,
start - 1);
                  fprintf (stderr, "Unable to
load seeds (%s)\n", fname);
                  exit (-1);
                  }
          }
for (c = (long) start; c <= children; c++)
          {
          for (i = PART; i <= PART_PSUPP; i++)
                  if (table & (1 << i))
                    {
                            if (verbose)
                                    fprintf
(stderr,

          "Generating seeds for %s [step: %d]  ",

          tdefs[i].comment, c);

                            step_size = scale /
children;
                            while (step_size >
MAX_32B_SCALE)
                                    {
                                    rowcnt =
tdefs[i].base * MAX_32B_SCALE;

          tdefs[i].gen_seed (rowcnt);
                                    step_size -=
MAX_32B_SCALE;
                                    }
                            rowcnt = tdefs[i].base
* step_size;
                            if (c == children)
                                    {
                                    int k, j=0;
```

```
                                    for (k = 0; k
< scale; k++)
                                            j +=
tdefs[i].base % children;
                                    rowcnt += j %
children;
                                    }
                            if (rowcnt > 1)

          tdefs[i].gen_seed (rowcnt);
                    }

            if (verbose)
                  fprintf (stderr, "\n");
            if (store_state (scale, children, c))
                  {
                    fprintf (stderr, "Unable to
store seeds\n");
                    exit (-1);
                  }
          }
  exit (0);
}

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

  table = (1 << CUST) |
         (1 << SUPP) |
         (1 << NATION) |
         (1 << REGION) |
         (1 << PART_PSUPP) |
         (1 << ORDER_LINE);
  force = 0;
  verbose = 0;
  columnar = 0;
  bld_seeds = 0;
  header = 0;
  oldtime = 0;
  direct = 0;
  scale = 1;
  flt_scale = 1.0;
  updates = 0;
  refresh = UPD_PCT;
  resume = -1;
  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;

#ifdef NO_SUPPORT
  signal (SIGINT, exit);
#endif /* NO_SUPPORT */
  process_options (ac, av);

#ifndef ORIG
    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 /* ORIG */

#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

  fprintf (stderr,
                        "TPC-D Population Generator
(Version %d.%d.%d%s)\n",
                        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)
         {
           for (i = LINE; i <= ORDER_LINE; i++)
                  {
                    if (table & (1 << i))
                           {
                             if (load_state
(scale, children, children))
                                    {
                                      fprintf
(stderr, "Unable to load seeds (%d)\n",

                                       scale);
                                      exit (-1);
                                    }
                             rowcnt =
tdefs[i].base / 10000 * scale * refresh;
                             upd_num = 0;
                             while (upd_num <
updates)
                                    {
```

```
                                        minrow =
rowcnt * upd_num + 1;
                                        if (verbose)

         fprintf (stderr,

                  "Generating updates for %s
[pid: %d]",

                  tdefs[i].comment, DSS_PROC);
                                        gen_tbl (i,
minrow, rowcnt, upd_num + 1);
                                        if (verbose)

         fprintf (stderr, "done.\n");
                                        pr_drange
(i, minrow, rowcnt, upd_num + 1);
                                        upd_num++;
                                      }
                          }
                 }
           exit (0);
         }
/*
 * seed gen is never parallelized
 */
/* The gen_seeds routine is new in v1.3.0 and it
does not work!!!
    if (bld_seeds)
        gen_seeds ((resume == -1) ? 1 : resume,
scale);
*/

/* It is replaced with the code from V1.2 */
    if (bld_seeds)
        {
                  if(resume==-1)
                          {
                          c=1;
                          }
        else
                          {
                          c=resume;
                          if (load_state(scale,
children, c-1))
                                  {
                                  char
fname[80];

         seed_name(fname, scale, children, c-1);

         fprintf(stderr, "Unable to load seeds
(%s)\n", fname);
                                  exit(-1);
                                  }
                          }
        for (; c <= children; c++)
            {
            for (i=PART; i <= PART_PSUPP; i++)
                if (table & (1 << i))
                    {
                    if ((tdefs[i].base * scale %
children) == 0)
                        rowcnt = tdefs[i].base *
scale / children;
                    else
                        {
                        fprintf(stderr, "Load
cannot be evenly split!");
                        exit(1);
                        }
                    if (verbose)
                        fprintf(stderr,
                            "Generating seeds for
%s [step: %d]  ",
                            tdefs[i].comment, c);
                    tdefs[i].gen_seed(rowcnt);
                    if (verbose)
                        fprintf(stderr,
"done.\n");
                    }
            if (store_state(scale, children, c))
                {
                fprintf(stderr, "Unable to store
seeds\n");
                exit(-1);
                }
            }
        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);
                          }
          }


  for (i = PART; i <= REGION; i++)
        if (table & (1 << i))
          {
                if (children > 1 && i < TIME)
                  if (step >= 0)
                        partial (i, step);
#ifdef DOS
                  else
                          {
                            fprintf (stderr,

"Parallel load is not supported on your
platform.\n");
                            exit (1);
                          }
#else
                  else
                          pload (i);
#endif /* DOS */
                else
                  {
                          minrow = 1;
                          if (i < TIME)
```

```
                            rowcnt =
tdefs[i].base * scale;
                          else
                            rowcnt =
tdefs[i].base;
                          if (verbose)
                            fprintf (stderr,
"Generating data for %s [pid: %d]",

tdefs[i].comment, DSS_PROC);
                          gen_tbl (i, minrow,
rowcnt, upd_num);
                          if (verbose)
                            fprintf (stderr,
"done.\n");
                  }
          }

  if (direct)
        close_direct ();

  return (0);
}
```

load_stub.c

```
/*************************************************
*****************
 *  Title:      load_stub.c
 *  Sccsid:     @(#)load_stub.c    9.1.1.11
9/7/95  16:24:05
 *  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"
#include "part.h"

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

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 *c, int mode)
{
  char line[256];
  char *ptr;

  if (partition) {
    partnum = find_partition(c->mktsegment, 1);
  }
  PPR_STRT(line, ptr);
  PPR_INT(ptr, c->custkey);
  PPR_STR(ptr, c->mktsegment, C_MSEG_LEN);
  PPR_INT(ptr, c->nation_code);
  PPR_VSTR(ptr, c->name, C_NAME_LEN);
  PPR_VSTR(ptr, c->address,
            (columnar)?(long)(ceil(C_ADDR_LEN *
V_STR_HGH)):c->alen);
  PPR_STR(ptr, c->phone, PHONE_LEN);
  PPR_MONEY(ptr, c->acctbal);
  PPR_VSTR(ptr, c->comment,
           (columnar)?(long)(ceil(C_CMNT_LEN *
V_STR_HGH)):c->clen);
  PPR_END(ptr);

  if (write(out[partnum], line, strlen(line)) ==
-1) {
    fprintf(stderr,"Error when writing customer
data\n");
    Exitprog(errno);
  }

  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)
{
  char line[256];
  char *ptr;
  double tmp = pp->size;

  if (partition) {
    partnum = find_partition(&(tmp), 0);
```

```
  }

  PPR_STRT(line, ptr);
  PPR_INT(ptr, pp->partkey);
  PPR_VSTR(ptr, pp->type,
           (columnar)?(long)P_TYPE_LEN:pp->tlen);
  PPR_INT(ptr, pp->size);
  PPR_STR(ptr, pp->brand, P_BRND_LEN);
  PPR_VSTR(ptr, pp->name,
           (columnar)?(long)P_NAME_LEN:pp->nlen);
  PPR_STR(ptr, pp->container, P_CNTR_LEN);
  PPR_STR(ptr, pp->mfgr, P_MFG_LEN);
  PPR_MONEY(ptr, pp->retailprice);
  PPR_VSTR(ptr, pp->comment,
           (columnar)?(long)(ceil(P_CMNT_LEN *
V_STR_HGH)):pp->clen);
  PPR_END(ptr);

  if (write(out[partnum], line, strlen(line)) ==
-1) {
    fprintf(stderr,"Error when writing parts
data\n");
    Exitprog(errno);
  }

  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_time (FILE *f)
{
    static int count = 0;

    if (! count++)
        printf("No header has been defined for
the time table\n");

    return(0);
}

int
ld_time (dss_time_t *tp, int mode)
{
    static int count = 0;

    if (! count++)
        printf("No load routine has been defined
for the time 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 *o, int mode)
{

  char line[256];
  char *ptr;
  static int last_mode = 0;

  if (mode != last_mode)
    last_mode = mode;

  if (partition) {
    (void) strcpy(tmpdate, o->odate);
    partnum = find_partition(tmpdate, 2);
  }
  PPR_STRT(line,ptr);
  PPR_STR(ptr, o->odate, DATE_LEN);
  PPR_BCD2(ptr, o->okey_high, o->okey_low);
  PPR_INT(ptr, o->custkey);
  PPR_STR(ptr, o->opriority, O_OPRIO_LEN);
  PPR_INT(ptr, o->spriority);
  PPR_STR(ptr, o->clerk, O_CLRK_LEN);
  PPR_CHR(ptr, o->orderstatus);
  PPR_MONEY(ptr, o->totalprice);
  PPR_VSTR(ptr, o->comment,
           (columnar)?(long)(ceil(O_CMNT_LEN *
V_STR_HGH)):o->clen);
  PPR_END(ptr);

  if(write(out[partnum], line, strlen(line)) == -
1) {

int
```

```
    fprintf(stderr,"Error when writing order
data\n");
    Exitprog(errno);
  }

  return(0);
}

ld_line (order_t *o, int mode)
{
  char line[256];
  char *ptr;
  static int last_mode = 0;
  long i;

  if ( mode != last_mode)
    last_mode = mode;

  for (i = 0; i < o->lines; i++) {
    if (partition) {
      (void) strcpy(tmpdate, o->l[i].sdate);
      partnum = find_partition(tmpdate, 2);
    }
    PPR_STRT(line, ptr);
    PPR_STR(ptr, o->l[i].sdate, DATE_LEN);
    PPR_BCD2(ptr, o->l[i].okey_high, o-
>l[i].okey_low);
    PPR_MONEY(ptr, o->l[i].discount);
    PPR_MONEY(ptr, o->l[i].eprice);
    PPR_INT(ptr, o->l[i].suppkey);
    PPR_INT(ptr, o->l[i].quantity);
    PPR_CHR(ptr, o->l[i].rflag[0]);
    PPR_INT(ptr, o->l[i].partkey);
    PPR_CHR(ptr, o->l[i].lstatus[0]);
    PPR_MONEY(ptr, o->l[i].tax);
    PPR_STR(ptr, o->l[i].cdate, DATE_LEN);
    PPR_STR(ptr, o->l[i].rdate, DATE_LEN);
    PPR_STR(ptr, o->l[i].shipmode, L_SMODE_LEN);
    PPR_INT(ptr, o->l[i].lcnt);
    PPR_STR(ptr, o->l[i].shipinstruct,
L_INST_LEN);
    PPR_VSTR(ptr, o->l[i].comment,
            (columnar)?(long)(ceil(L_CMNT_LEN *
V_STR_HGH)):o->l[i].clen);
    PPR_END(ptr);
    if(write(out[(partnum)], line, strlen(line))
== -1) {
      fprintf(stderr,"Error when writing lineitem
data\n");
      Exitprog(errno);
    }
  }


  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);
}
```

part.c

```
/*=============================================
==================+
 |       Copyright (c) 1997  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
 |    fjlee     06/02/97 - add shared permission
for open

+=============================================
==================*/

#include "part.h"

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

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 = (int *) malloc(numpart * sizeof(int));

  /* 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) ==
NULL) {
        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] = open(ofile,
O_WRONLY|O_APPEND|O_CREAT,S_IWRITE|S_IREAD);
    if (out[j] == -1) {
      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;

  /* 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++)
    close(out[i]);

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


int find_partition(cval,type)
     void *cval;
     int type;
{
  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:
    ASS_DATE(vdat,(char *)cval);
    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++;
  }
}
```

part.h

```
/*=============================================
====================+
 |       Copyright (c) 1997  Oracle Corp, Redwood
Shores, CA        |
 |                    ADVANCED TECH PERFORMANCE
GROUP                  |
 |                         All Rights Reserved
|

+============================================
==================+
 | FILENAME
 |    part.h
 | DESCRIPTION
 |    Header file for dbgen partitioning
extension.
 | MODIFIED
 |    pswong    03/13/97 - created

+============================================
==================*/

#include <windows.h>
#include <winbase.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <share.h>

#define FLEN 128
```

```
#define BLEN 10240

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

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

typedef struct ot {
  HANDLE    hfile;
  int             olen;
  char   obuf[BLEN];
  HANDLE    omutex;
} otype;

#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, "%51d%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();
```

Appendix C. Query Text and Query Output

qual_query.log

```
Begin Execution at Fri Feb 05 19:15:50 1999


-- using default substitutions
-- Query 1 (Original)

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            3773034.00
5319329289.68    5053976845.78
5256336547.68    25.51            35964.01
0.05             147907.00
N            F            100245.00
141459686.10     134380852.77
139710306.87     25.63            36160.45
0.05             3912.00
N            O            7464940.00
10518546073.98   9992072944.46
10392414192.06   25.54            35990.13
0.05             292262.00
R            F            3779140.00
5328886172.99    5062370635.93
5265431221.82    25.55            36025.46
0.05             147920.00


4 rows processed.
Statement Processed in 0.38 seconds.
Setting the number of rows to fetch to: 100


Ended Executing this Query at Fri Feb 05 19:15:50
1999


Query Started at 147350.25
Query Ended at 147350.63
Query Processed in 0.38 seconds.

------------------------------------------------

-- Query 2 (Variant A)


Began Executing this Query at Fri Feb 05 19:15:50
1999


SELECT
S_ACCTBAL,
S_NAME,
N_NAME,
P_PARTKEY,
P_MFGR,
S_ADDRESS,
S_PHONE,
S_COMMENT
FROM PARTS, 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 (P_PARTKEY ,PS_SUPPLYCOST) IN
(SELECT PS_PARTKEY, MIN(PS_SUPPLYCOST)
FROM PARTSUPP, SUPPLIER, NATION, REGION
WHERE S_SUPPKEY = PS_SUPPKEY
AND S_NATIONKEY = N_NATIONKEY
AND N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'EUROPE'
GROUP BY PS_PARTKEY)
ORDER BY S_ACCTBAL DESC, N_NAME, S_NAME,
P_PARTKEY

S_ACCTBAL        S_NAME                    N_NAME
P_PARTKEY        P_MFGR
S_ADDRESS                                  S_PHONE
S_COMMENT
9828.21          Supplier#000000647        UNITED
KINGDOM
13120.00         Manufacturer#5
jB16PyPyB7Bl52jMjSPw3mS                    33-258-
202-4782
z1QhSiMj11Bm7COlLwh6Ql0B1R2Mg4CLn
LhiP0wiMzy72hlkP7l5in2y6RS6N130lz51nSRL5gOg5S26hP
CCQN2L
9508.37          Supplier#000000070        FRANCE
3563.00          Manufacturer#1
M5C616R5h5SlMR3zzmLkSw24j2                 16-821-
608-1166
m7z0CPShmBkhlChBAi3LkQ2CLw
mhl6QP362RPS3044CB2y41yhOhjlBin0CL7yhxmhS4hBM07kQ
1yyjOjz3C
9508.37          Supplier#000000070        FRANCE
17268.00         Manufacturer#4
M5C616R5h5SlMR3zzmLkSw24j2                 16-821-
608-1166
m7z0CPShmBkhlChBAi3LkQ2CLw
mhl6QP362RPS3044CB2y41yhOhjlBin0CL7yhxmhS4hBM07kQ
1yyjOjz3C
9453.01          Supplier#000000802
ROMANIA
10021.00         Manufacturer#5
```

5yARQNSLNRAlOlBnkNQCik3SOlyClk7nmRhA2h0 29-342-882-6463
65y3RQ2i0OP6Nz7mS hC
9453.01 Supplier#000000802 ROMANIA
13275.00 Manufacturer#4
5yARQNSLNRAlOlBnkNQCik3SOlyClk7nmRhA2h0 29-342-882-6463
65y3RQ2i0OP6Nz7mS hC
9192.10 Supplier#000000115 UNITED KINGDOM
13325.00 Manufacturer#1
h0m3lzlSPMw2B0ny7LNyNCkjRRn7iyMlLBLA 33-597-248-1220
1QzQjhSyx
9032.15 Supplier#000000959 GERMANY
4958.00 Manufacturer#4
205LNCzxMCnQ5gnz4n S3ynP6Mhnw 17-108-642-3106
Px z7kOx56l7jQz NwBBQhky yM7kLgxRQw5zw6
8702.02 Supplier#000000333 RUSSIA
11810.00 Manufacturer#3
5iwkgN5n2BN15OmQk2602h0N6NzxPyiPN5lnj 32-508-202-6136
SgimAjmn3wL7Rlxmh3LCwOPnhjyl 7xxzxAN
8615.50 Supplier#000000812 FRANCE
10551.00 Manufacturer#2
h4i2M2O0 ky1g2mlBOmxjzj0hA2h6nkSNhP 16-585-724-6633
57i0NAyR0RP2jOh54C6B22OlSL
8615.50 Supplier#000000812 FRANCE
13811.00 Manufacturer#4
h4i2M2O0 ky1g2mlBOmxjzj0hA2h6nkSNhP 16-585-724-6633
57i0NAyR0RP2jOh54C6B22OlSL
8488.53 Supplier#000000367 RUSSIA
6854.00 Manufacturer#4
nkmQ2Qzgh0wA 3x Sn2S7N5gmSOj xwC COSn6 32-458-198-9557
35C2RROP C Nlgi2N
8430.52 Supplier#000000646 FRANCE
11384.00 Manufacturer#3
61SjP6S y B0 32l11 16-601-220-5489
kiw4NSNBNxy5kywzwyx0PMM21xiMOhxR423Akkm
8271.39 Supplier#000000146 RUSSIA
4637.00 Manufacturer#5
wh yPSk6hNBlB4l33iQ0wS0 RhBhQ4zQ3lz 32-792-619-3155
jjwgwljRO63 n7OM2MP0hg3L1mlwBMLmMlS4Cgyn
8096.98 Supplier#000000574 RUSSIA
323.00 Manufacturer#4
hCOj4Cgx43xx jgP4QkL7gLN65 32-866-246-8752
OhxNj6SlB56315B3k5SCBzwQyLk76zlj4Ow2Q
BC2wACkxh3S0RCyx6nARzSQR2010k0BCPhOg6yQm
7392.78 Supplier#000000170 UNITED KINGDOM
7655.00 Manufacturer#2
PCxjjzNQihLNxgLw0SiMmQ 33-803-340-5398
M116S1xzg54iC3k7OPLQi3Cimhghz2BClQk
g5Ag12QSBlhglANnw4MR MBS 72A
7205.20 Supplier#000000477 GERMANY
10956.00 Manufacturer#5
Mimj6403h zmAzAgg Bjy05O 2z 17-180-144-7991
yRlyR SnMxmhPjAmBw
S02AxQ6yOhBRlOwzmlxzO0A2Sx075kjlAknn7z2
O0S7hy0BiknwOQm6Pmz3gL4gj2z7
6820.35 Supplier#000000007 UNITED KINGDOM
13217.00 Manufacturer#5
z45m2jBRzl5ilLNz4 33-990-965-2201
1PhngjmiSQl0RzRACP0l4S70xSL
QPSBM16072SkMLCgm4OOMjARLNQk3g1P3BB32AgBMl462B0CP7Rh24
6721.70 Supplier#000000954 FRANCE
4191.00 Manufacturer#3
OM7xnNxNnkgQ mzh2g3RQmg1g 16-537-341-8517
5ni3yCkmz5ymx0kCg74zhLA B516Si1w152AkiByx1Nl
NgghAkkmNz1jASj4mxzxznOySg7hAyM3MRRnBj
6329.90 Supplier#000000996 GERMANY
10735.00 Manufacturer#2
k6135gA3zPwNl7L3Rl45mlnACjngOQQBB300iyA 17-447-811-3282
PBO7wjlQMm1h3AAA 1NQAl0kkijnkRNgQ0 mh1z6QS0gC5lP1
ykmzNR20OlN506ARS0z3j
6173.87 Supplier#000000408 RUSSIA
18139.00 Manufacturer#1
Cni6 zR5C4lh104POx5h05 mg53CQ2Sw4SAM2M2x 32-858-724-2950
10SxMOwhjON3khzQ124gNnyw7B4nL7ml4L5llSR
5364.99 Supplier#000000785 RUSSIA
13784.00 Manufacturer#4
71OnPzQkC2P1hRNRgjyQP4n1 32-297-653-2203
kiiPQ3ik7R ykAhRx43Rw70L1Ok
7AMi3AjRw7lklwxwyiL6S2O1COyS4QB46m5M167mjMwCm0w
5069.27 Supplier#000000328 GERMANY
16327.00 Manufacturer#1
5O4033xSgml 17-231-513-5721
OMk3ALAPNmj6lBLMAS7M1nCAS
4xLj51iy2klix3nPi26gAxPgANmk6zSi6 3A7m
l11BOwiC6xLB4hBRiPM
4941.88 Supplier#000000321 ROMANIA
7320.00 Manufacturer#5
hyLQ mg42S2kAMlj M3BwMSjS 29-573-279-1406
y2644kMhOkPCm5P5y7Lmz7OR6mgSmBN631RggmC
4672.25 Supplier#000000239 RUSSIA
12238.00 Manufacturer#1
y4ymj7B5BN1nMSkwPPggAl 32-396-654-6826
Py3RA2gykmSCmj0z3ii7Rxzhz6OyR RxS C3S23LPQ
4586.49 Supplier#000000680 RUSSIA
5679.00 Manufacturer#3
BP1Nlw5nPMxRnOAwm 32-522-

```
382-1620
kA0y25RNO1Al im7SyiPzSym3M5OS52l6S576kn0S2k
0mPBLlAzL6Ax7CM6iNi4CgCy6Bln7hlhxmlRng
4518.31          Supplier#000000149        FRANCE
18344.00         Manufacturer#5
4B QSy5Bl2                                 16-660-
553-2456
hijkPhgL1g4L1Q27y0Q42wh0Qz3jPiL4NgkM4NNg1
llQ1yNNBk1C1QnlRO7 4ki
4315.15          Supplier#000000509        FRANCE
18972.00         Manufacturer#2
B5 iPRn7L4yMllgwCnRPMA                     16-298-
154-3365
ygiPh7ymP7jBznmR2lQLLgjmi1wik
3526.53          Supplier#000000553        FRANCE
8036.00          Manufacturer#4
ylLOx2gMw 5iBl6AiNL60Q                     16-599-
552-3755
L3ggShlRlyxmR4MNl7Rw7OQign6yO
3526.53          Supplier#000000553        FRANCE
17018.00         Manufacturer#3
ylLOx2gMw 5iBl6AiNL60Q                     16-599-
552-3755
L3ggShlRlyxmR4MNl7Rw7OQign6yO
3294.68          Supplier#000000350
GERMANY
4841.00          Manufacturer#4
x5kRL2z1BPg0 BO 2hi1iOyh 30RRg0OPj         17-113-
181-4017
BjQznni44OmQ7S16yl3zxk2M6nM4M 27yMPML
2972.26          Supplier#000000016        RUSSIA
1015.00          Manufacturer#4
B7wLkSLRjNS MS1C                           32-822-
502-4215
C7w6S6QzhAPQmMmNmMN1hA0llQOA 00m1NmC25wyQ461SA
jy03zmRh22MLM00zhmi
2963.09          Supplier#000000840
ROMANIA
3080.00          Manufacturer#2
1ynwiQkNh0 CMRRCk4l306M2ij0jykg6QNgSCAzy 29-781-
337-5584
S7NRMx43RmOjxML6hxLyN75LzxBwB0wjSLx3
S3Cwh52S6ilSOLhQm0 6C1 yzx3jPm6Sjg
5By0BCPwOR32i1CQgxR0gB43gh
2221.25          Supplier#000000771
ROMANIA
13981.00         Manufacturer#2
LAjCRjl3nAMzzhmw0Sx1Mg                     29-986-
304-9006
jhk0N7NlhS23iCngC52BBC
0jilM0wByx0LB5R070R2lCx113lQiS7xNhBRA0xknlNxLiA
1381.97          Supplier#000000104        FRANCE
18103.00         Manufacturer#3
i Qnl4 1 jiwM C2yxAylL5R4SBQh54N6           16-434-
972-6922
MwnBw1g71Pig2Am7nz0Mm5SNl7OwQLAkN56ji
906.07           Supplier#000000138
ROMANIA
8363.00          Manufacturer#4
1iBxSxLl1Mh3 6LS6PlLPNlnlMjCQh22z6n5        29-533-
434-6776
nLjQAmCw77R2jRMgz5LSyyx1QN l
4jMMO3RAkxOkzRmwQl3Qm5236k72RRPnim0 BkzQnBMM6A
PMml2n
765.69           Supplier#000000799        RUSSIA
11276.00         Manufacturer#2
Am7yihz47mg NkgQL w By4                    32-579-
339-1495
MMRPNQ 4l66mQQPNniAiiL0PQ2C4yyBRn1nRlxnxkj5Ak45Pw
mQk1ROhz66BRQiiL gPRQRy 56MyQ nS1N14R
7Ml6xhl2lOS3
727.89           Supplier#000000470
ROMANIA
6213.00          Manufacturer#3
gAySBM2N7 PgwP5kiP4n7BzOik0M               29-165-
289-1523
zCkPgn
6wN5A3R47gljlQ3hNSLShP2RALxCiinkOy4wCwA1LCiBO5yiS
C yBAA lii
683.07           Supplier#000000651        RUSSIA
4888.00          Manufacturer#4
ymQ6PByCh4lzxBBPLB2wwOhRh47wQMQSPL         32-181-
426-4490
kx6jhQkwzg6RkRgPLPM30BgL1R726l1m5AMk0MmMQBQ
nCihiLxhMgCgRih6MmMx0PglRQ7AQnl72g50
167.56           Supplier#000000290        FRANCE
2037.00          Manufacturer#1
xm6n30QmBn75QPh7Py01lxlnB4n1lNh  MBi1      16-675-
286-5102
gi6y41Bg5
AkhmCh30Qyxx515xLi6MRjCQ4n2xjML7N0PNSCyBPwS1C
91.39            Supplier#000000949        UNITED
KINGDOM
9430.00          Manufacturer#2
zCmkwm43LQ62Q3O 4nkNA                      33-332-
697-2768
Nm5yn23SxgPAMRz6B5CjAj04nkNx5lA7O
-314.06          Supplier#000000510
ROMANIA
17242.00         Manufacturer#4
hwx1kP6nQ0NgPO232nxAwL4QPjRN i64jOiPjC0    29-207-
852-3454
5NAyRmwSCQ5hMB6RiN0Qyhjghnl0BPjPlNlP725AAOCw
-820.89          Supplier#000000409
GERMANY
2156.00          Manufacturer#5
Nk4z1lg25gNwMwO 2BnMOn1P3k0yA7i2l          17-719-
517-9836
CmLnngQywQ4hhx3266yz5Qgj
nOm1P1RC3gxmg4kmnNCSnM4PBO4hNQA1xyzC2LCQBP07w2z3
-845.44          Supplier#000000704
ROMANIA
9926.00          Manufacturer#5
zi jjmO4A57 mQgLmhALCM3B zL                29-300-
896-5991
OSh0PRm6ByyS2wLBNPQPPi2CL0zN0Px1LNOlwl6y0Rgij
zP5hPCz 745nn62hPORl6i7SynBCC4Qy4Qkn1 l
-942.73          Supplier#000000563
GERMANY
5797.00          Manufacturer#1
xSLLy 4jm17OC                              17-108-
537-2691
1gSOl3A0jnkAmQLllmygzmg7hhjjAmRxN5SwjxM16Ok61
wm5N42


44 rows processed.
Statement Processed in 3.63 seconds.
Setting the number of rows to fetch to: 10


Ended Executing this Query at Fri Feb 05 19:15:54
1999


Query Started at 147350.63
Query Ended at 147354.26
Query Processed in 3.63 seconds.
```

```
------------------------------------------------

-- Query 3 (Original)


Began Executing this Query at Fri Feb 05 19:15:54
1999


SELECT
L_ORDERKEY,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE,
TO_CHAR(O_ORDERDATE, 'YYYY-MM-DD'),
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

L_ORDERKEY       REVENUE
TO_CHAR(O_ORDERD O_SHIPPRIORITY
260930.00        320547.25        1995-03-12
0.00
402497.00        298879.53        1995-02-12
0.00
457859.00        296490.68        1995-01-17
0.00
509889.00        294068.87        1995-02-03
0.00
58117.00         292632.83        1995-02-21
0.00
538311.00        279666.00        1995-03-07
0.00
588421.00        275477.12        1995-03-03
0.00
416167.00        273765.45        1995-02-22
0.00
97830.00         273227.06        1995-03-04
0.00
90276.00         272233.92        1995-03-04
0.00


10 rows processed.
Statement Processed in 3.68 seconds.

Ended Executing this Query at Fri Feb 05 19:15:58
1999


Query Started at 147354.26
Query Ended at 147357.94
Query Processed in 3.68 seconds.

------------------------------------------------

-- Query 4 (Original)


Began Executing this Query at Fri Feb 05 19:15:58
1999


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        999.00
2-HIGH          1002.00
3-MEDIUM        1021.00
4-NOT SPECIFIED 997.00
5-LOW           1089.00


5 rows processed.
Statement Processed in 2.04 seconds.

Ended Executing this Query at Fri Feb 05 19:16:00
1999


Query Started at 147357.94
Query Ended at 147359.98
Query Processed in 2.04 seconds.

------------------------------------------------

-- Query 5 (Original)


Began Executing this Query at Fri Feb 05 19:16:00
1999


SELECT
N_NAME,
SUM(L_EXTENDEDPRICE*(1-L_DISCOUNT)) AS REVENUE
FROM CUSTOMER, ORDERS, LINEITEM, SUPPLIER,
NATION, REGION
WHERE C_CUSTKEY = O_CUSTKEY
AND O_ORDERKEY = L_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
CHINA                     7349391.47
INDONESIA                 6485853.40
INDIA                     5505346.82
JAPAN                     5388883.59
```

```
VIETNAM                   4728846.60


5 rows processed.
Statement Processed in 0.72 seconds.

Ended Executing this Query at Fri Feb 05 19:16:00
1999


Query Started at 147359.98
Query Ended at 147360.70
Query Processed in 0.72 seconds.

------------------------------------------------

-- Query 6 (Original)


Began Executing this Query at Fri Feb 05 19:16:00
1999


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
11450588.04


1 row  processed.
Statement Processed in 0.12 seconds.

Ended Executing this Query at Fri Feb 05 19:16:00
1999


Query Started at 147360.70
Query Ended at 147360.82
Query Processed in 0.12 seconds.

------------------------------------------------

-- Query 7 (Original)


Began Executing this Query at Fri Feb 05 19:16:00
1999


SELECT
SUPP_NATION,
CUST_NATION,
YEAR,
SUM(VOLUME) AS REVENUE
FROM
(SELECT
N1.N_NAME AS SUPP_NATION,
N2.N_NAME AS CUST_NATION,
TO_CHAR(L_SHIPDATE,'YYYY') AS 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, YEAR
ORDER BY SUPP_NATION, CUST_NATION, YEAR

SUPP_NATION               CUST_NATION
YEAR REVENUE
FRANCE                    GERMANY
1995 4611421.44
FRANCE                    GERMANY
1996 4828420.37
GERMANY                   FRANCE
1995 6755766.84
GERMANY                   FRANCE
1996 5810951.40


4 rows processed.
Statement Processed in 0.56 seconds.

Ended Executing this Query at Fri Feb 05 19:16:01
1999


Query Started at 147360.82
Query Ended at 147361.38
Query Processed in 0.56 seconds.

------------------------------------------------

-- Query 8 (Variant B)


Began Executing this Query at Fri Feb 05 19:16:01
1999


SELECT
YEAR,
SUM(DECODE(NATION, 'BRAZIL', VOLUME, 0)) /
SUM(VOLUME) AS MKT_SHARE
FROM
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE*(1-L_DISCOUNT) AS VOLUME,
N2.N_NAME AS NATION
FROM PARTS, 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 YEAR
ORDER BY YEAR

YEAR MKT_SHARE
1995 0.05
1996 0.09


2 rows processed.
Statement Processed in 8.23 seconds.

Ended Executing this Query at Fri Feb 05 19:16:09
1999


Query Started at 147361.38
Query Ended at 147369.61
Query Processed in 8.23 seconds.

------------------------------------------------

-- Query 9 (Original)


Began Executing this Query at Fri Feb 05 19:16:09
1999



SELECT
NATION,
YEAR,
SUM(AMOUNT) AS SUM_PROFIT
FROM
(SELECT
N_NAME AS NATION,
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) - PS_SUPPLYCOST
* L_QUANTITY AS AMOUNT
FROM PARTS, 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, YEAR
ORDER BY NATION, YEAR DESC

NATION                    YEAR SUM_PROFIT
ALGERIA                   1998 1946316.01
ALGERIA                   1997 2973825.69
ALGERIA                   1996 3308881.52
ALGERIA                   1995 3092227.30
ALGERIA                   1994 3406958.71
ALGERIA                   1993 3140744.03
ALGERIA                   1992 3330704.41
ARGENTINA                 1998 3045410.01
ARGENTINA                 1997 4255378.59
ARGENTINA                 1996 4651751.94
ARGENTINA                 1995 4897797.00
ARGENTINA                 1994 4823465.77
ARGENTINA                 1993 4499810.71
ARGENTINA                 1992 4764593.39
BRAZIL                    1998 2932051.36
BRAZIL                    1997 3784531.35
BRAZIL                    1996 3965665.69
BRAZIL                    1995 4063060.86
BRAZIL                    1994 4236277.35
BRAZIL                    1993 4363461.31
BRAZIL                    1992 4684749.23
CANADA                    1998 2217064.04
CANADA                    1997 2950110.61
CANADA                    1996 3184049.97
CANADA                    1995 3962540.19
CANADA                    1994 3365251.02
CANADA                    1993 3617013.37
CANADA                    1992 3407955.25
CHINA                     1998 3048192.02
CHINA                     1997 5001207.69
CHINA                     1996 4800958.31
CHINA                     1995 5154927.73
CHINA                     1994 5882634.53
CHINA                     1993 4733364.82
CHINA                     1992 5014704.08
EGYPT                     1998 1892538.74
EGYPT                     1997 3849220.07
EGYPT                     1996 3418656.55
EGYPT                     1995 3766170.60
EGYPT                     1994 3520025.56
EGYPT                     1993 4375424.75
EGYPT                     1992 4586034.39
ETHIOPIA                  1998 1860117.73
ETHIOPIA                  1997 3705722.33
ETHIOPIA                  1996 3577215.39
ETHIOPIA                  1995 3425219.55
ETHIOPIA                  1994 3428616.18
ETHIOPIA                  1993 3459815.43
ETHIOPIA                  1992 3280072.91
FRANCE                    1998 1592531.55
FRANCE                    1997 2746176.54
FRANCE                    1996 2505844.88
FRANCE                    1995 2902077.00
FRANCE                    1994 2532229.56
FRANCE                    1993 2305725.44
FRANCE                    1992 2955126.69
GERMANY                   1998 3538625.73
GERMANY                   1997 4425943.40
GERMANY                   1996 4266344.96
GERMANY                   1995 3952963.52
GERMANY                   1994 4462655.80
GERMANY                   1993 4435094.66
GERMANY                   1992 4521715.41
INDIA                     1998 3378369.34
INDIA                     1997 4186477.85
INDIA                     1996 5074383.92
INDIA                     1995 4487435.38
INDIA                     1994 4718312.63
INDIA                     1993 4499573.81
INDIA                     1992 4712930.33
INDONESIA                 1998 2902077.10
INDONESIA                 1997 4973644.23
INDONESIA                 1996 4977652.49
INDONESIA                 1995 5359380.15
INDONESIA                 1994 4854637.20
INDONESIA                 1993 4213131.42
INDONESIA                 1992 4999478.51
IRAN                      1998 2415763.10
IRAN                      1997 4227175.11
IRAN                      1996 4527365.03
IRAN                      1995 4139514.72
IRAN                      1994 4166316.39
```

```
IRAN                      1993 3366959.59
IRAN                      1992 3599399.70
IRAQ                      1998 2596922.63
IRAQ                      1997 3707054.11
IRAQ                      1996 3726138.38
IRAQ                      1995 4350503.89
IRAQ                      1994 4131512.79
IRAQ                      1993 3787196.42
IRAQ                      1992 4043738.13
JAPAN                     1998 2265666.94
JAPAN                     1997 3988819.28
JAPAN                     1996 4319004.53
JAPAN                     1995 4262698.64
JAPAN                     1994 3545212.62
JAPAN                     1993 4051565.97
JAPAN                     1992 3692137.45
JORDAN                    1998 1978591.74
JORDAN                    1997 3315454.29
JORDAN                    1996 3236531.98
JORDAN                    1995 2778207.99
JORDAN                    1994 2420301.07
JORDAN                    1993 3272130.93
JORDAN                    1992 2649126.09
KENYA                     1998 2265677.73
KENYA                     1997 3493019.32
KENYA                     1996 3346373.30
KENYA                     1995 3537360.32
KENYA                     1994 2800950.72
KENYA                     1993 3477468.30
KENYA                     1992 2719618.04
MOROCCO                   1998 2549499.93
MOROCCO                   1997 3891824.90
MOROCCO                   1996 3730777.74
MOROCCO                   1995 3469641.13
MOROCCO                   1994 3747593.21
MOROCCO                   1993 3620742.70
MOROCCO                   1992 4303609.25
MOZAMBIQUE                1998 2024719.46
MOZAMBIQUE                1997 3706003.09
MOZAMBIQUE                1996 3376430.93
MOZAMBIQUE                1995 2737631.64
MOZAMBIQUE                1994 3373146.48
MOZAMBIQUE                1993 3608300.37
MOZAMBIQUE                1992 3551263.95
PERU                      1998 2142791.97
PERU                      1997 4664076.15
PERU                      1996 3623628.93
PERU                      1995 3908939.79
PERU                      1994 3386204.16
PERU                      1993 3877048.49
PERU                      1992 3768394.25
ROMANIA                   1998 1760625.70
ROMANIA                   1997 2707685.33
ROMANIA                   1996 2553345.48
ROMANIA                   1995 2715901.59
ROMANIA                   1994 3023644.06
ROMANIA                   1993 2873247.32
ROMANIA                   1992 2728060.71
RUSSIA                    1998 2975973.22
RUSSIA                    1997 3785806.47
RUSSIA                    1996 4217625.59
RUSSIA                    1995 3883445.52
RUSSIA                    1994 4395855.01
RUSSIA                    1993 3900944.18
RUSSIA                    1992 4691358.61
SAUDI ARABIA              1998 2931482.83
SAUDI ARABIA              1997 5498943.16
SAUDI ARABIA              1996 4473723.74
SAUDI ARABIA              1995 5939212.93
SAUDI ARABIA              1994 4527695.71
SAUDI ARABIA              1993 4928702.02
SAUDI ARABIA              1992 5527261.52
UNITED KINGDOM            1998 3198731.37
UNITED KINGDOM            1997 4363882.74
UNITED KINGDOM            1996 4730956.67
UNITED KINGDOM            1995 4842014.55
UNITED KINGDOM            1994 4912706.56
UNITED KINGDOM            1993 4415255.96
UNITED KINGDOM            1992 4375524.23
UNITED STATES             1998 1892045.16
UNITED STATES             1997 3102027.86
UNITED STATES             1996 3334320.26
UNITED STATES             1995 3168244.60
UNITED STATES             1994 3296960.10
UNITED STATES             1993 3558109.05
UNITED STATES             1992 2755129.39
VIETNAM                   1998 2906627.03
VIETNAM                   1997 4544560.45
VIETNAM                   1996 4314259.00
VIETNAM                   1995 4365340.86
VIETNAM                   1994 3686987.71
VIETNAM                   1993 3764237.18
VIETNAM                   1992 3420922.00

175 rows processed.
Statement Processed in 4.98 seconds.
Setting the number of rows to fetch to: 20

Ended Executing this Query at Fri Feb 05 19:16:14
1999

Query Started at 147369.61
Query Ended at 147374.59
Query Processed in 4.98 seconds.

-------------------------------------------------

-- Query 10 (Original)

Began Executing this Query at Fri Feb 05 19:16:14
1999

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
```

```
C_CUSTKEY        C_NAME
REVENUE          C_ACCTBAL
N_NAME                    C_ADDRESS
C_PHONE
C_COMMENT
9722.00          Customer#000009722
464618.26        474.04
CANADA                    l Mwzn4NAk6j
13-518-602-8070
5L 50Oy
RSgBAzPxmOSi5wk6xxOR7kh2nnPlgy7LBng2hOw5B01
RmCM120L24Pkg7PS1zwC11BCnz4L6i15PkixP26166
12800.00         Customer#000012800
444265.64        1900.84
PERU
57zjB3CQx4P4OB2R2MBi2mwhSllM4mn 4 nC6    27-142-
205-3552
0hwglS77RB56Rx436lQ0N16CxhOPnmyhgwz
5z64wnj1kiC4jL350mM41y71hNxBllPjyA4hiN1wzjjM7SCxA
N244mk2A
1025.00          Customer#000001025
442028.02        3363.46
INDIA                     lkiSn154M5ROi
18-588-456-4616
0Bl45z233Rniw0OO064nPBgP16kimO0y74iLh73g1N4 m310
jQ yQzPA50iC 3MA75g2Bj162Nw4P
13028.00         Customer#000013028
441692.24        -452.66
UNITED KINGDOM            yP714ORSNgNN2LA3L5B
33-253-660-2127
xPkmnhL2BkhkNyww4khlxwwAymN
h11PSjBCNMi50LkyOhO6CC 5nzOQCALzliOk2R66w
l05hRPO3iSP
3694.00          Customer#000003694
438180.07        2960.44
UNITED KINGDOM            2CCklmCBOCC
33-421-331-3127
MzLxQxLlLx3MPxlAwg1B5kg61zxkPnk
xiAm6PhMMAAQ2nzN3S6zzgP
x70w0lhhPx4QRzlMMy0204lAl3mBO7jh2jAP0N60wg367z
976.00           Customer#000000976
435897.63        7772.85
ROMANIA                    QzR 56Px1kgS
wANnAz02RS 30n Pm         29-436-660-4732
kzn32776 gwzkMzzzO4yxOAnkR7hR4R4x2SMwilz3x6h
nN7OnNLRMml3 kz5SLwi1yklOxiwS4g0wmA5A
4hmgBSwRRiQ1
8206.00          Customer#000008206
429905.11        6046.36
ARGENTINA                 P yMg30BBBBx
NMgC03AmzN2                 11-571-859-1370
hLi122RMPmLC36Oy0kxO71zz2wCR0QQCl7z26hlQ3mM
13532.00         Customer#000013532
427731.80        -924.18
KENYA
6ij7M5PBMx2kwwyz62Oj4SL5S0mRCwl3m1Rmw    24-525-
332-7244
7ih7yRz214zO67AiNPx64nO5l5k
yj6i3jLA5PCL15Q4QlA3ll60iM1P iBxCixg6
1hCh2RCnjOzk5R OnO 1OhhC3m4631m5
12745.00         Customer#000012745
422327.69        9691.33
CHINA                     SgS1LMC4gB2NM3wh
28-985-189-6174
jl72wjSw0 S6 7L4Cgxw PkyO5Nl2LL7LBR
2344.00          Customer#000002344
411240.11        5597.22
MOROCCO                   O3PC7ikBgw OAzPAlm2P
426zm3BnBN6Q1O 6N   25-593-745-7663
5NBn0wRNngLw2z5kyn1AhL0ASyg6SMhM
i2kMOyxARAnlO0Q5j4CBNARix7ABlMAC
2656.00          Customer#000002656
401185.95        8115.55
ALGERIA                   On55lAS3Rm5RxS m
10-667-469-8092
46ABx4jgni mlBMPCLxRhyPQM4RNS 5yO1L7zSOmk
MhPxAxQQ6lQnLj
17LymOhi4l5innzOyB2Olxzmw3gmx0SxiyBN5CSMNgCkLCkMg
O
59.00            Customer#000000059
400759.15        3458.60
ARGENTINA                 wP6CMyClly0lS4CAM1mzm
11-355-584-3112
l1g7xBCxxC7SM
5AkmnmAk00677O1MzA2R7A0Cx0Njixj56jL2iN
PNkSNQiy55m6ki3OgnhM47mSR7B
7069.00          Customer#000007069
396217.52        8198.94
INDONESIA
55Cw7ChL4Bi5ONn2A4m2i2n4nSNQQMjml        19-644-
744-1798
6jNS624l75zlxNli4lxO5zyPykPS1xnilS0NhkgOAkSx7P
6553.00          Customer#000006553
385863.59        8985.90
MOZAMBIQUE                R3LnnxONBjCLC0MRkxy7
26-166-724-4677
S7CkNLwA3kh006j71lwAlC25Bw6AMQ6i
6C0OSS6O7ARNNny60Ogh  3642mRxyiAgy5yk
3nPO4473wkNg5R6gzO4lz3zmM2m7MiLAilCC
3095.00          Customer#000003095
384246.11        8829.21
IRAQ                      S1gMCnBLwzi mCgB664
jl00Ll1Snh1iPMgCgR5  21-847-218-8188
3LSx7PxS
A4A5Cl3gAy3mg4Qj2xQlyx7xM1kA664AM7zmMmzORh3C1h
MO3nw6MymiljAMg65hOMB4Sn44kO w0lin7
3391.00          Customer#000003391
382541.78        7742.35
CANADA                    m3 CORmQNLzkShymLS
iMkCimRSl20 NB        13-592-494-2668
ynMlmhMBA5ikC1nCghlmAhQ0 675S3y2R33yjkNPQOS
13678.00         Customer#000013678
376280.56        9030.40
MOROCCO
BMk77lQm1lwNA0LghAkg3hCwNl4              25-306-
951-3937
mOS55RASx1wP136nQ5xBLznLhgw1kQ6PO6imNxQ7kR0x71P0S
zByMzh
6062.00          Customer#000006062
374512.65        1370.35
CANADA
n5zzil60zyxAlkzx7x1nihigPzR OBkR znMOMh  13-756-
700-4918
4zAm4wNB
li4QRPgPz2wM541x043hmLj4O3LBkALCP16hj2RQBO1OMNly7
ww1QP7w5i SSn0jNhAR yQmmz1hi5j3
554.00           Customer#000000554
373004.47        8395.57
BRAZIL                    jC5zhQky4zQB27lB5Sm
AQhQ  Px0            12-938-503-7317
0nxCl3 xSmiLQO 1M
2n0NCiRlnMMxP25j26x2igLhNOxjgMgmwwy7OkjzCACOg0z2L
AjOm0RPRmOPiCAAQwLlQSg 1yS3 gLCM1M2BzjnSjPl3nwAkk
13126.00         Customer#000013126
371722.00        6172.91
INDIA
xPAS4MnPh40i5Q2h4NQ61zz4RkyAwANA        18-288-
```

```
190-4145
nniMkAN6C0ClQ0mMmPz27liz4hk6L
2MlwPxh42N1l0R2hRwxxzlwMkxO4MAyz7RCj43NxLwQ3m6P27
yAj


20 rows processed.
Statement Processed in 7.59 seconds.

Ended Executing this Query at Fri Feb 05 19:16:22
1999


Query Started at 147374.59
Query Ended at 147382.18
Query Processed in 7.59 seconds.

------------------------------------------------

-- Query 11 (Original)


Began Executing this Query at Fri Feb 05 19:16:22
1999


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
12098.00         16227681.21
5134.00          15709338.52
13334.00         15023662.41
17052.00         14351644.20
3452.00          14070870.14
12552.00         13332469.18
1084.00          13170428.29
5797.00          13038622.72
12633.00         12892561.61
403.00           12856217.34

[ rows removed ]

14628.00         1016033.20
10126.00         1015846.78
3884.00          1014413.50
16913.00         1013604.40
18644.00         1010288.10
19870.00         1007919.36
18564.00         1007416.20
10179.00         1004920.00
883.00           1004650.68
3627.00          1004461.04

2541 rows processed.
Statement Processed in 4.90 seconds.

Ended Executing this Query at Fri Feb 05 19:16:27
1999


Query Started at 147382.18
Query Ended at 147387.08
Query Processed in 4.90 seconds.

------------------------------------------------

-- Query 12 (Variant B)


Began Executing this Query at Fri Feb 05 19:16:27
1999


SELECT
L_SHIPMODE,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',1,'2-
HIGH',1,0)) AS HIGH_LINE_COUNT,
SUM(DECODE(O_ORDERPRIORITY,'1-URGENT',0,'2-
HIGH',0,1)) 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       654.00           950.00
SHIP       684.00           1004.00


2 rows processed.
Statement Processed in 1.98 seconds.

Ended Executing this Query at Fri Feb 05 19:16:29
1999


Query Started at 147387.08
Query Ended at 147389.06
Query Processed in 1.98 seconds.

------------------------------------------------

-- Query 13 (Original)


Began Executing this Query at Fri Feb 05 19:16:29
1999


SELECT
YEAR,
SUM(REVENUE) AS REVENUE
FROM
```

```
(SELECT
TO_CHAR(O_ORDERDATE,'YYYY') AS YEAR,
L_EXTENDEDPRICE * (1-L_DISCOUNT) AS REVENUE
FROM LINEITEM, ORDERS
WHERE  O_ORDERKEY = L_ORDERKEY
AND O_CLERK = 'Clerk#000000088'
AND L_RETURNFLAG = 'R'
) PERFORMANCE
GROUP BY YEAR
ORDER BY YEAR

YEAR REVENUE
1992 1262855.73
1993 964121.03
1994 1750395.29
1995 198820.30


4 rows processed.
Statement Processed in 2.17 seconds.

Ended Executing this Query at Fri Feb 05 19:16:31
1999


Query Started at 147389.06
Query Ended at 147391.23
Query Processed in 2.17 seconds.

------------------------------------------------

-- Query 14 (Variant C)


Began Executing this Query at Fri Feb 05 19:16:31
1999


CREATE TABLE ALL_SALES0 (TYPE VARCHAR2(25),
AMOUNT NUMBER(20, 2))
Statement Processed in 0.12 seconds.

CREATE TABLE SUM_PROMO_SALES0 (PROMO_AMOUNT
NUMBER(20, 2))
Statement Processed in 0.02 seconds.

CREATE TABLE SUM_ALL_SALES0 (ALL_AMOUNT
NUMBER(20, 2))
Statement Processed in 0.02 seconds.

INSERT INTO ALL_SALES0
SELECT P_TYPE, SUM(L_EXTENDEDPRICE *(1-
L_DISCOUNT))
FROM  LINEITEM, PARTS
WHERE L_PARTKEY = P_PARTKEY
AND L_SHIPDATE >= TO_DATE('1995-09-01','YYYY-MM-
DD')
AND L_SHIPDATE < ADD_MONTHS(TO_DATE('1995-09-
01','YYYY-MM-DD'), 1)
GROUP BY P_TYPE
Statement Processed in 4.01 seconds.

commit
Statement Processed in 0.00 seconds.

INSERT INTO SUM_PROMO_SALES0
SELECT SUM(AMOUNT)
FROM  ALL_SALES0
WHERE TYPE LIKE 'PROMO%'
Statement Processed in 0.00 seconds.

commit
Statement Processed in 0.00 seconds.

INSERT INTO SUM_ALL_SALES0
SELECT SUM(AMOUNT)
FROM ALL_SALES0
Statement Processed in 0.00 seconds.

commit
Statement Processed in 0.00 seconds.

SELECT 100.00*PROMO_AMOUNT/ALL_AMOUNT AS
PROMO_REVENUE
FROM SUM_PROMO_SALES0, SUM_ALL_SALES0

PROMO_REVENUE
16.73


1 row  processed.
Statement Processed in 0.00 seconds.

commit
Statement Processed in 0.00 seconds.

DROP TABLE ALL_SALES0
Statement Processed in 0.18 seconds.

DROP TABLE SUM_PROMO_SALES0
Statement Processed in 0.02 seconds.

DROP TABLE SUM_ALL_SALES0
Statement Processed in 0.02 seconds.

Ended Executing this Query at Fri Feb 05 19:16:35
1999


Query Started at 147391.23
Query Ended at 147395.65
Query Processed in 4.42 seconds.

------------------------------------------------

-- Query 15 (Variant B)


Began Executing this Query at Fri Feb 05 19:16:35
1999


CREATE TABLE REVENUE0 (SUPPLIER_NO NUMBER,
TOTAL_REVENUE NUMBER(20, 2))
Statement Processed in 0.02 seconds.

INSERT INTO REVENUE0
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 1.86 seconds.

commit
```

```
Statement Processed in 0.00 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
389.00           Supplier#000000389
PB1Lx0xx6LMz3h7Rx63m6j3QmMx              34-885-
883-5717 1418538.21


1 row  processed.
Statement Processed in 0.01 seconds.

DROP TABLE REVENUE0
Statement Processed in 0.02 seconds.

Ended Executing this Query at Fri Feb 05 19:16:37
1999


Query Started at 147395.65
Query Ended at 147397.57
Query Processed in 1.92 seconds.

------------------------------------------------

-- Query 16 (Original)


Began Executing this Query at Fri Feb 05 19:16:37
1999


SELECT
P_BRAND,
P_TYPE,
P_SIZE,
COUNT(DISTINCT PS_SUPPKEY) AS SUPPLIER_CNT
FROM PARTSUPP,PARTS
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 '%Better Business
Bureau%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#14   SMALL ANODIZED NICKEL     45.00
12.00
Brand#22   SMALL BURNISHED BRASS     19.00
12.00
Brand#25   PROMO POLISHED COPPER     14.00
12.00
Brand#35   LARGE ANODIZED STEEL      45.00
12.00
Brand#35   PROMO BRUSHED COPPER      9.00
12.00
Brand#51   ECONOMY ANODIZED STEEL    9.00
12.00
Brand#53   LARGE BRUSHED NICKEL      45.00
12.00
Brand#11   ECONOMY POLISHED COPPER   14.00
8.00
Brand#11   LARGE PLATED STEEL        23.00
8.00
Brand#11   PROMO POLISHED STEEL      23.00
8.00

[ rows removed ]

Brand#23   SMALL ANODIZED BRASS      23.00
3.00
Brand#24   PROMO BRUSHED STEEL       49.00
3.00
Brand#31   SMALL PLATED NICKEL       36.00
3.00
Brand#32   LARGE BRUSHED STEEL       45.00
3.00
Brand#41   PROMO BURNISHED NICKEL    36.00
3.00
Brand#43   STANDARD BRUSHED BRASS    23.00
3.00
Brand#44   MEDIUM ANODIZED NICKEL    9.00
3.00
Brand#51   MEDIUM BRUSHED TIN        36.00
3.00
Brand#53   MEDIUM BURNISHED BRASS    49.00
3.00
Brand#54   SMALL POLISHED BRASS      9.00
3.00


2762 rows processed.
Statement Processed in 2.85 seconds.

Ended Executing this Query at Fri Feb 05 19:16:40
1999


Query Started at 147397.57
Query Ended at 147400.42
Query Processed in 2.85 seconds.

------------------------------------------------

-- Query 17 (Original)


Began Executing this Query at Fri Feb 05 19:16:40
1999


SELECT
SUM(L_EXTENDEDPRICE)/7.0 AS AVG_YEARLY
FROM LINEITEM, PARTS
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
```

```
24436.88


1 row  processed.
Statement Processed in 8.75 seconds.

Ended Executing this Query at Fri Feb 05 19:16:49
1999


Query Started at 147400.42
Query Ended at 147409.17
Query Processed in 8.75 seconds


SQL statements processed: 34
Queries processed: 17
```

Appendix D. Seed and Query Substitution Parameters

Seed

- Using 454986351 as a seed to the RNG.

Substitution Parameters

- Query 1

  72

- Query 4

  1994-01-01

- Query 15b

  1994-01-01

- Query 10

  1993-06-01

- Query 11

  ETHIOPIA 0.0000010000

- Query 6

  1994-01-01 0.04 24

- Query 2a

  11 COPPER AMERICA

- Query 16

  Brand#22 MEDIUM ANODIZED 47 2 42 39 32 21 13

  5

- Query 14c

  1994-02-01

- Query 8b

  ETHIOPIA AFRICA SMALL PLATED COPPER

- Query 12b

  AIR RAIL 1997-01-01

- Query 17

  Brand#22 LG CAN

- Query 3

  BUILDING 1995-03-09

- Query 5

  AMERICA 1994-01-01

- Query 13

  Clerk#000000206

- Query 7

  ETHIOPIA FRANCE

- Query 9

  cornsilk

Appendix E. Implementation Specific Layer/Driver Code

gettime.c

```
/* Copyright (c) Oracle Corporation 1995.  All
Rights Reserved. */
/*
NAME    GETTIME.C

DESCRIPTION

NOTES
   Desktop Performance Group

MODIFIED       (MM/DD/YY)
   B Moriarty   06/02/95 -  V4.8 Initial Version

*/

#include <windows.h>
#include <sys\types.h>
#include <time.h>
#include <stdio.h>

# ifdef __STDC__
double gettime(void)
# else
double gettime()
# endif /* __STDC__ */
{

   LARGE_INTEGER PFreq;             /* Ticks per
Second */
   LARGE_INTEGER PCount;   /* Ticks Since 1970 */
   double count;
   double freq;

   if (!QueryPerformanceFrequency(&PFreq))
      return(-1.);
   if (!QueryPerformanceCounter(&PCount))
      return(-1.);
   freq = (double)PFreq.QuadPart;
   count = (double)PCount.QuadPart;
   return(count/freq);

}
```

qexecpl.c

```
/*=============================================
==================+
 |       Copyright (c) 1997  Oracle Corp, Redwood
Shores, CA        |
 |                    OPEN SYSTEMS PERFORMANCE
GROUP                 |
 |                         All Rights Reserved
|

+=============================================
==================+
 | FILENAME
 |    qexecpl.c
 | DESCRIPTION
 |    SQL Execution Engine, Oracle v8,  OCI
version
 | MODIFIED
 |    pswong    02/19/97 - migrating to version 8
 |    pswong    04/02/96 - more polishing
 |    pswong    03/25/96 - polish up
 |    pswong      03/06/96 - created

+=============================================
==================*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();
extern char *getenv();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

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

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

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

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

/* For our purpose of timing, we will treat
comments as delimiters */
```

```
/* for queries.   Thus, we will collect query
timings whenever we  */
/* encounter a comment (of course not for the
first comment in a   */
/* file).
*/

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

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

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

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

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

char stmt[SQL_LEN];       /* The SQL statement or
comment line.  */
char cmnt[81];            /* Buffer to save the
comment.         */

FILE *qtemp = stdin;      /* fd for query
template               */
FILE *logfile = stdout;   /* log and report files
*/
FILE *rep = stdout;

void *defbuf;             /* Buffer pointer for
ODEFIN           */
int deflen = 0;           /* Size of data type
for ODEFIN        */
int deftype = 1;          /* Oracle type number
for ODEFIN       */

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

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

/* OCI handles */

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

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

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

void usage() {

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



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

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

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

```
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
    if (type)
      (void)
OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)ms
g,
                           2048,OCI_HTYPE_ERROR);
    else
      (void)
OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)ms
g,
                           2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }

  /* Rollback just in case */

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

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

  SQLexit();

  exit(1);
}


void main(argc,argv)
     int argc;
     char *argv[];
{

  int i;
  int retcode;      /* Return code for
get_statement */

  /* Initialize some variables */

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

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

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

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

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

  SQLinit();

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

  time(&tim);
  fprintf(logfile, "Begin Execution at %s\n\n",
ctime(&tim));

  /* Get the next statement and start processing
it */

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

    switch (retcode) {

      /* If this is a comment, skips it */
    case COMMENT:
      if (end_flag) {
        end_flag = 0;      /* reset query end
flag */
        /* save the comment so that we can print
it out later on  */
        strcpy(cmnt, stmt);
        break;
      }
      fprintf(logfile, "%s", stmt);
      fprintf(rep, "%s", stmt);
      break;

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

      /* if this is a SQL statement */
```

```
    case SQL_STMT:

      /* Executes the query */

      SQLexec();

      s_tr_end = gettime();
      stmt_cnt++;

      /*
      fprintf(logfile,"\nStatement Started at
%.2f\n", s_tr_start);
      fprintf(logfile,"Statement Ended at
%.2f\n", s_tr_end);
      */
      fprintf(logfile,"Statement Processed in
%.2f seconds.\n",
               (s_tr_end - s_tr_start));
      break;

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

  /* Get Timing for the last query */

  tr_end = gettime();

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

  fprintf(rep, "%.2f\n", (tr_end - tr_start));

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

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}



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

void SQLinit()
{
  int i;
  char *pddl_degree;
  char *pdml_degree;

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

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

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

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

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

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

  /* get username and password */

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

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR
_SERVER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(l
ogname),OCI_ATTR_USERNAME,
           errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(pa
sswd),OCI_ATTR_PASSWORD,
           errhp);
```

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR
_SESSION,errhp);

  /* Enable session parallel dml */

  if (getenv((const char *) "PDML_DEGREE") !=
(char *)0)
    pdml_degree = getenv((const char *)
"PDML_DEGREE");
  else
    pdml_degree = "80";

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

  if (getenv((const char *) "PDDL_DEGREE") !=
(char *)0)
    pddl_degree = getenv((const char *)
"PDDL_DEGREE");
  else
    pddl_degree = "8";

  sprintf((char *) stmt, PDDLTXT, pddl_degree);

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

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

}



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

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

  if (!end_flag) {

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

    tr_end = gettime();

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

      fprintf(logfile,"-------------------------
----------------------\n\n");

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

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

      fprintf(rep, "%.2f\n", (tr_end -
tr_start));
      fprintf(rep, "%s", cmnt);

      fprintf(logfile,"\nBegan Executing this
Query at %s\n", ctime(&tim));

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

      fflush(logfile);
      fflush(rep);

    }

    tr_start = tr_end;
    qry_cnt++;

    end_flag = 1;
  }

  s_tr_start = gettime();

  /* prepare the statement */

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

  /* Prints the query text to the logfile */
```

```
  fprintf(logfile, "\n%s\n", stmt);

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


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATT
R_STMT_TYPE,errhp);

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

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

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

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

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

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


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_AT
TR_ROW_COUNT,errhp);

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

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

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

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

  /* free all memory */

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

  /* Flush all output */

  fflush(rep);
  fflush(logfile);

}


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

int define_output_variables()
{

  int i;
  int retflag = 0;

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

    slist[i].buflen = MAX_COLNAME_SIZE;

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


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

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

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

/*
```

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

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

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

    switch (slist[i].dbtype) {

    case OCI_TYPECODE_NUMBER:

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

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

      break;

    default:

      /* default is character string */

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

    /* Define the column */

    if ((status=OCIDefineByPos(curq,&(dlist[i]-
>defhdl),errhp,POS(i),

defbuf,deflen,deftype,NULL,
                                    dlist[i]-
>rlen,NULL,OCI_DEFAULT))!=OCI_SUCCESS)
      sql_error(errhp,status,1);
  }
  return i;
}
```

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

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

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

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

  print_header(num);

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

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

  /* Fetch the rows and print them out */

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

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


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_AT
TR_ROW_COUNT,errhp);

    print_rows(num,rows_ret);

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

    if (stats != OCI_NO_DATA) {
      if (num_to_fetch == -1) {
         while ((stats =
OCIStmtFetch(curq,errhp,MAX_ARRAY,OCI_FETCH_NEXT,

OCI_DEFAULT)) == OCI_SUCCESS) {

OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                        OCI_ATTR_ROW_COUNT,errhp);
           print_rows(num,(num_so_far-rows_ret));
           rows_ret = num_so_far;
         }
         /* Print the final rows */
         OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,
NULL,
                      OCI_ATTR_ROW_COUNT,errhp);
         print_rows(num,(num_so_far-rows_ret));
         rows_ret = num_so_far;
      } else {
         ntf -= MAX_ARRAY;

         while ((stats = OCIStmtFetch(curq,errhp,

((ntf>MAX_ARRAY) ? MAX_ARRAY:ntf),

OCI_FETCH_NEXT, OCI_DEFAULT)) ==
                OCI_SUCCESS) {
           ntf -= MAX_ARRAY;
```

```
OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                       OCI_ATTR_ROW_COUNT,errhp);
          print_rows(num,(num_so_far-rows_ret));
          rows_ret = num_so_far;
          if (ntf <= 0) break;
        }
        OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,
NULL,
                      OCI_ATTR_ROW_COUNT,errhp);
        print_rows(num,(num_so_far-rows_ret));
        rows_ret = num_so_far;
      }
    }
  } else {
    OCIStmtFetch(curq, errhp, ntf,
OCI_FETCH_NEXT, OCI_DEFAULT);

OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_AT
TR_ROW_COUNT,errhp);
    print_rows(num,rows_ret);
  }

  fprintf(logfile,"\n\n%d row%c processed.\n",
rows_ret,
          rows_ret == 1 ? '\0' : 's');

}


int get_statement()
{

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

  /* Reset statement buffer */

  stmt[0] = '\0';

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

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

    /* remove blanks */

    str = line;

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

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

    strcat(stmt, str);

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

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

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


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

void *memalloc(size)
     int size;
{

  void *tmp;

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



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

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

  fprintf(logfile, "\n");

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

    /* extract the column name */

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

    /* format the output a little */

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

    /* do a little bit of formatting */

    if (cwid > 80) {
      fprintf(logfile,"\n");
      len = 0;
    } else if ((len += cwid) > 80) {
      fprintf(logfile,"\n");
      len = cwid;
    }
#ifdef FORMAT1
    if ((slist[i].dbtype == INT_TYPE) ||
(slist[i].dbtype == FLT_TYPE))
```

```
      fprintf(logfile, "%*s ", cwid,
slist[i].buf);
    else  /* string type */
      fprintf(logfile, "%*s ", -cwid,
slist[i].buf);
#else
      fprintf(logfile, "%*s ", -cwid, colname);
#endif /* FORMAT1 */
  }

  fprintf(logfile,"\n");
}


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

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

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

    len = 0;

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

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

      /* do a little bit of formatting */

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

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

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

void remove_newline(str)
     char *str;
{

  char *p;

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

qexecpl.h

```
/*================================================
====================+
 |       Copyright (c) 1995  Oracle Corp, Redwood
Shores, CA        |
 |                    OPEN SYSTEMS PERFORMANCE
GROUP                  |
 |                        All Rights Reserved
|

+================================================
====================+
 | FILENAME
 |    qstream.h
 | DESCRIPTION
 |    SQL statement execution front-end header
file.
 | MODIFIED
 |    pswong    03/07/96 - created

+================================================
====================*/

#ifndef QSTREAMPL_H

#define QSTREAMPL_H

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

#include <oratypes.h>

#include <oratypes.h>

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

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

/* some basic definitions */
```

```
#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

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

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */

#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)

/* defines and typedefs for parsing */

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

/* defines and typedefs for query description */

#define MAX_COLNAME_SIZE 256   /* Maximum length
of Column name  */
#define MAX_SEL_LIST 16       /* Maximum items on
a select list */

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

/* types for describe */

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

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

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

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

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

#define MAX_ARRAY 1000       /* Maximum array
size for array fetch */
#define PFMEMSIZE 65536    /* Memory size of
prefetch buffer     */

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

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

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

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

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

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

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

/* For get_statement */
```

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


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

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

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

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

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

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


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

#endif /* QSTREAMPL_H */
```

resuf1.ksh

```
#!/bin/ksh
#
#================================================
==================+
#        Copyright (c) 1995  Oracle Corp, Redwood
Shores, CA       |
#                   OPEN SYSTEM PERFORMANCE GROUP
|
#                        All Rights Reserved
|
#================================================
==================+
# FILENAME
#    resuf1.ksh
# DESCRIPTION
#    resuf1.ksh [-u <uid/passwd to login>]
#               <set id> <parallelism> <scale
factor>
# USAGE
#    To restore UF1.
#================================================
==================
TPCD_DIR=e:/tpdc814_work
SCRIPT_DIR=${TPCD_DIR}/scripts
SQL_DIR=${TPCD_DIR}/sql
LOAD_CTL=${TPCD_DIR}/loadctl/update

usage() {
echo ""
echo "resuf1.ksh [-u <user/passwd>]"
echo "           [-h] <set id> <parallelism>
<scale factor>"
echo "set_id       : the set ID that the update
flatfiles belong"
echo "parallelism  : The parallelism of the
updates."
echo "scale factor : Scale Factor of the
database."
echo ""
echo "-u           : Userid/Password for Oracle.
Default is tpcd/tpcd"
echo "-h           : To Display this message."
echo ""
}

PASSWD="tpcd/tpcd"

set -- `getopt "u: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

SET_ID=$1
PAR=$2
SF=$3


if [ ${SF} -eq 0 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
LOG_DIR=${TPCD_DIR}/log/tenthgig
DBF_DIR=\\TPCD\\DBS\\TENTHGIG
fi

if [ ${SF} -eq 1 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onegig
LOG_DIR=${TPCD_DIR}/log/onegig
DBF_DIR=\\TPCD\\DBS\\ONEGIG
```

```
fi

if [ ${SF} -eq 3 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threegig
LOG_DIR=${TPCD_DIR}/log/threegig
DBF_DIR=\\TPCD\\DBS\\THREEGIG
fi

if [ ${SF} -eq 10 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/tengig
LOG_DIR=${TPCD_DIR}/log/tengig
DBF_DIR=\\TPCD\\DBS\\TENGIG
fi

if [ ${SF} -eq 30 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
LOG_DIR=${TPCD_DIR}/log/thirtygig
DBF_DIR=\\TPCD\\DBS\\THIRTYGIG
fi

if [ ${SF} -eq 100 ]
then
UPDATE_DIR=${TPCD_DIR}\\update\\data\\hundgig
LOG_DIR=${TPCD_DIR}\\log\\hundgig
DBF_DIR=\\TPCD\\DBS\\HUNDGIG
fi

if [ ${SF} -eq 300 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
LOG_DIR=${TPCD_DIR}/log/threehundgig
DBF_DIR=\\TPCD\\DBS\\THREEHUNDGIG
fi

if [ ${SF} -eq 1000 ]
then
UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
LOG_DIR=${TPCD_DIR}/log/onetbyte
DBF_DIR=\\TPCD\\DBS\\ONETBYTE
fi

i=1
PID=""

# perform the reset function 1

echo "Update Function 2 (resuf1) Set $SET_ID
started!"

START=`gtime`

# first create the temp tables

sqlplus ${PASSWD}
@e:/tpcd814_work/sql/build_temp_okey1.sql

# All processes have started, now wait for all
the update processes to finish.

wait

while [ $i -le $PAR ]
do

  SQLLDR userid=$PASSWD
control=e:/tpcd814_work/loadctl/update/tempokey1.
ctl \
  log=e:/tpcd814_work/log/hundgig/ti${i}_ruf1.log
\

data=e:/tpcd814_work/update/data/hundgig/okeyuf1.
u${SET_ID}.${i} \
  direct=true parallel=true &

  i=`expr $i + 1`

done

wait

sqlplus $PASSWD <<!

spool e:\tpcd814_work\log\hundgig\resuf1.log;
set timing on;
analyze table temp_okey1 estimate statistics
sample 1000 rows;

alter session force parallel dml;

delete /*+ ordered use_nl(lineitem)
index(lineitem,l_ored) */
from lineitem where l_orderkey in
(select /*+ nl_sj */ t_orderkey from temp_okey1);

delete /*+ ordered use_nl(orders)
index(orders,o_okey) */
from orders where o_orderkey in
(select /*+ nl_sj */ t_orderkey from temp_okey1);

commit;

drop table temp_okey1;
spool off;
exit;
!

END=`gtime`

# Done

echo ""
echo "Update Function 2 (resuf1) Set $SET_ID
done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

resuf2.ksh

```
#!/bin/ksh
#
#================================================
==================+
#        Copyright (c) 1995  Oracle Corp, Redwood
Shores, CA       |
#                   OPEN SYSTEM PERFORMANCE GROUP
|
#                         All Rights Reserved
|
#================================================
==================+
# FILENAME
#    resuf2.ksh
# DESCRIPTION
#    resuf2.ksh -u [<uid/passwd>]
#              <set id> <parallelism> <scale
```

```
factor>
#
# USAGE
#    To execute UF1.
#================================================
==================
TPCD_DIR=e:/tpcd814_work
SCRIPT_DIR=${TPCD_DIR}/scripts
SQL_DIR=${TPCD_DIR}/sql
LOAD_CTL=e:\\tpcd814_work\\loadctl\\update

usage() {
echo ""
echo "resuf2.ksh -u [<uid/passwd>]"
echo "           [-h] <set id> <parallelism>
<scale factor>"
echo ""
echo "set_id       : the set ID that the update
flatfiles belong"
echo "parallelism  : The parallelism of the
updates."
echo "scale factor : Scale Factor of the
database."
echo ""
echo "-u           : Userid/Password for Oracle.
Default is tpcd/tpcd."
echo "-h           : To Display this message."
echo ""
}

PASSWD="tpcd/tpcd"

set -- `getopt "l:u: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

SET_ID=$1
PAR=$2
SF=$3

if [ ${SF} -eq 0 ]
then
CSF=.1
UPDATE_DIR=${TPCD_DIR}/update/data/tenthgig
LOG_DIR=${TPCD_DIR}/log/tenthgig
DBF_DIR=\\TPCD\\DBS\\TENTHGIG
fi

if [ ${SF} -eq 1 ]
then
CSF=1
UPDATE_DIR=${TPCD_DIR}/update/data/onegig
LOG_DIR=${TPCD_DIR}/log/onegig
DBF_DIR=\\TPCD\\DBS\\ONEGIG
fi

if [ ${SF} -eq 3 ]
then
CSF=3
UPDATE_DIR=${TPCD_DIR}/update/data/threegig
LOG_DIR=${TPCD_DIR}/log/threegig
DBF_DIR=\\TPCD\\DBS\\THREEGIG
fi

if [ ${SF} -eq 10 ]
then
CSF=10
UPDATE_DIR=${TPCD_DIR}/update/data/tengig
LOG_DIR=${TPCD_DIR}/log/tengig
DBF_DIR=\\TPCD\\DBS\\TENGIG
fi

if [ ${SF} -eq 30 ]
then
CSF=30
UPDATE_DIR=${TPCD_DIR}/update/data/thirtygig
LOG_DIR=${TPCD_DIR}/log/thirtygig
DBF_DIR=\\TPCD\\DBS\\THIRTYGIG
fi

if [ ${SF} -eq 100 ]
then
CSF=100
UPDATE_DIR=${TPCD_DIR}/update/data/hundgig
#UPDATE_DIR=i:/flatfiles
LOG_DIR=e:\\tpcd814_work\\log\\hundgig
DBF_DIR=\\TPCD\\DBS\\HUNDGIG
fi

if [ ${SF} -eq 300 ]
then
CSF=300
UPDATE_DIR=${TPCD_DIR}/update/data/threehundgig
LOG_DIR=${TPCD_DIR}\\log/threehundgig
DBF_DIR=\\TPCD\\DBS\\THREEHUNDGIG
fi

if [ ${SF} -eq 1000 ]
then
CSF=1000
UPDATE_DIR=${TPCD_DIR}/update/data/onetbyte
LOG_DIR=${TPCD_DIR}/log/onetbyte
DBF_DIR=\\TPCD\\DBS\\ONETBYTE
fi

i=1
PID=""

# Let's figure out the extent sizes in Mb
LINEEXT=`echo $CSF '* 720 * 0.001 / ' $PAR + 1 |
bc`
LINEEXT=`echo $LINEEXT | sed
'/\.[0123456789]*/s///'`
ORDEXT=`echo $CSF '* 200 * 0.001 / ' $PAR + 1 |
bc`
ORDEXT=`echo $ORDEXT | sed
'/\.[0123456789]*/s///'`

# perform the reset function 2

echo "Update Function 2 (resuf2) Set $SET_ID
started!"

START=`gtime`
```

```
# first create the temp tables

sqlplus ${PASSWD}
@e:/tpcd814_work/sql/build_temp_ordline.sql
${ORDEXT}m ${LINEEXT}m

while [ $i -le $PAR ]
do

# Kick off sqlldr to load the data into the
temporary staging tables.

   SQLLDR userid=$PASSWD
control=${LOAD_CTL}/tempord.ctl \

log=e:/tpcd814_work/log/hundgig/to${i}_ruf2.log \

data=e:/tpcd814_work/update/data/hundgig/rorder.t
bl.u${SET_ID}.${i} \
   direct=true parallel=true &

   SQLLDR userid=$PASSWD
control=${LOAD_CTL}/tempitem.ctl \

log=e:/tpcd814_work/log/hundgig/ti${i}_ruf1.log \

data=e:/tpcd814_work/update/data/hundgig/rlineite
m.tbl.u${SET_ID}.${i} \
   direct=true parallel=true &

   i=`expr $i + 1`

done

# All sqlldr processes have started, now wait for
all them to finish.

wait

# now do the insert and then drop the staging
tables

sqlplus $PASSWD <<!

spool $LOG_DIR/resuf2.log
set timing on;
alter session force parallel dml;

analyze table temp_ord estimate statistics sample
1000 rows;
analyze table temp_item estimate statistics
sample 1000 rows;

insert into orders (select * from temp_ord);

insert into lineitem (select * from temp_item);

commit;

drop table temp_item;
drop table temp_ord;
spool off;
exit;
!

END=`gtime`

# Done

echo ""
echo "Update Function 1 (resuf2) Set $SET_ID
done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

audit.sh

```
#!/bin/ksh
#
================================================
=================+
#         Copyright (c) 1996  Oracle Corp,
Redwood Shores, CA       |
#                    OPEN SYSTEMS PERFORMANCE
DIVISION               |
#                        All Rights Reserved
|
#
================================================
=================+
#  FILENAME
#     audit.sh
#  DESCRIPTION
#     Usage: audit.sh [<all|first|second>]
#
#     Run multiple stream audit
#
================================================
=================+

#
# Get params
#
FIRST="yes"
SECOND="yes"

if [ "$1" = "first" ]; then
  SECOND="no"
else
  if [ "$1" = "second" ]; then
    FIRST="no"
  fi
fi

#
# Set PDML/PDDL degree
#

# update 1:00 21 jan
#export PDML_DEGREE=80
export PDML_DEGREE=16
export PDDL_DEGREE=10

if [ "$FIRST" = "yes" ]; then
  #
  # Bounce database before first run
  #
#  tshut
#  sleep 60
#  tstart

  echo "Starting first run"
  date

  #
  # first run
  #
```

```
  ./audit_multi.sh 100 first

  echo "Done with first run"
  date

  echo
  echo
fi


if [ "$SECOND" = "yes" ]; then
  #
  # Bounce database before second run
  #
#  tshut
#  sleep 60
#  tstart

  echo "Starting second run"
  date

  #
  # second run
  #
  ./audit_multi.sh 100 second

  echo "Done with second run"
  date
fi
```

audit_multi.sh

```
#!/bin/ksh
#
================================================
=================+
#         Copyright (c) 1996  Oracle Corp,
Redwood Shores, CA      |
#                   OPEN SYSTEMS PERFORMANCE
DIVISION              |
#                       All Rights Reserved
|
#
================================================
=================+
#  FILENAME
#     audit_multi.sh
#  DESCRIPTION
#     Usage: audit_multi.sh <scale_factor>
<first|second>
#
#     Run multiple stream audit
#
================================================
=================+
TPCD_HOME=e:/tpcd814_work
SCRIPT_DIR=${TPCD_HOME}/scripts
SQL_DIR=${TPCD_HOME}/sql
UPD_DIR=${TPCD_HOME}/update
SRC_DIR=${TPCD_HOME}/source
QRY_DIR=${TPCD_HOME}/queries
QGEN_DIR=${TPCD_HOME}/bumpx/dbgen
QGEN=${QGEN_DIR}/qgen
TPCD_DIR=$TPCD_HOME #modeified at 20,jan

DSS_QUERY=${TPCD_HOME}/Dqueries
export DSS_QUERY

RUN_ID_FILE=${TPCD_HOME}/r_id

UPD_SQL=${UPD_DIR}/sql
UPD_SPT=${UPD_DIR}/scripts
UPD_SRC=${UPD_DIR}/source
UPD_DAT=${UPD_DIR}/data

TPCD_BIN=${TPCD_HOME}/multistream
TPCD_LOG=${TPCD_HOME}/log
TPCD_RPT=${TPCD_HOME}/rpt

OUT=${TPCD_HOME}/out
GTIME=${SRC_DIR}/gtime
HID=1

# The defaults
USER="tpcd/tpcd"
QPROG=${TPCD_HOME}/bin/qexec.exe

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 ""
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=2
  START_SET_UPDATE=2
  STOP_SET_UPDATE=3
else
  UF_SET=4
  START_SET=1
  STOP_SET=2
  START_SET_UPDATE=5
  STOP_SET_UPDATE=6
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`
export RUN_ID
echo $RUN_ID > $RUN_ID_FILE

echo "TPC-D Power Test Run `date`"
echo "RUNID is $RUN_ID"
echo ""

echo "TPC-D 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 for stream0"
echo "Generates query template file for stream0"
>> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
########
 ${QGEN} -c -p 0 -s ${SF} > ${QRY_FILE}
########
echo "Done generating query template file for
stream0"
echo "Done generating query template file for
stream0" >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

START=`$GTIME`
echo "TPC-D Power Test Execution starts at
$START"
echo "TPC-D 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 at ${SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

UF1_START=`${GTIME}`
echo "Start Time for UF1 is $UF1_START"
echo "Start Time for UF1 is $UF1_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

########
 ${SCRIPT_DIR}/audit_uf1.sh $UF_SET >
${OUT}/uf1.${RUN_ID}.${HID} 2>&1
########

# Execute Query Stream

UF1_END=`${GTIME}`
echo "End Time for UF1 is $UF1_END"
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}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "Start Query\t${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}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "Start UF2 at ${SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
UF2_START=`${GTIME}`
echo "Start Time for UF2 is $UF2_START"
echo "Start Time for UF2 is $UF2_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

########
 ${SCRIPT_DIR}/audit_uf2.sh $UF_SET >
${OUT}/uf2.${RUN_ID}.${HID} 2>&1
########

UF2_END=`${GTIME}`
echo "End Time for UF2 is $UF2_END"
echo "End Time for UF2 is $UF2_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
EDATE=`date`
echo "END UF2 at ${EDATE}"
echo "End UF2 at ${EDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

END=`$GTIME`

echo "TPC-D Power Test Execution ends at $END"
echo "TPC-D Power Test Execution ends at $END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
MEA_INT=`echo $END - $START | bc`
echo "Measurement Interval is $MEA_INT"
echo " " >> ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "Power Measurement Interval is $MEA_INT" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt


UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo "Elapsed Time for Update Function 1 is
```

```
$UF1_TIME"
echo "Elapsed Time for Update Function 1 is
$UF1_TIME" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

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 2 is
$UF2_TIME"
echo "Elapsed Time for Update Function 2 is
$UF2_TIME" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

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 17 \* 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

echo ""
echo "-----------------------------------------
----"
echo ""

i=$START_SET
while [ $i -le $STOP_SET ]; do

TPCD_LOG_FILE=${TPCD_LOG}/query.${RUN_ID}_${i}.lo
g

TPCD_RPT_FILE=${TPCD_RPT}/query.${RUN_ID}_${i}.rp
t
  QRY_FILE=${QRY_DIR}/qtemp.${RUN_ID}_${i}.${SF}

  echo "Generates query template files for STREAM
${i}"

########
# ${QGEN} -c -p ${i} -s ${SF} > ${QRY_FILE}
########

 i=`expr $i + 1`
 done

echo "Done Qgen for multi-stream"
echo

#
# Starting multiple streams run
#
echo ""
echo "-----------------------------------------
----"
echo ""
M_START=`$GTIME`
echo "Multi-stream starts at $M_START"
echo "Multi-stream starts at $M_START" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

# rm -f /tmp/th_pipe
# mknod /tmp/th_pipe p

#
# Starting all the query streams
#
i=$START_SET
while [ $i -le $STOP_SET ]; do
  echo "Start Query\t${M_SDATE}" >>
${TPCD_RPT}/tpcd.${RUN_ID}_${i}.${HID}.rpt
  echo "Start Query\t${M_SDATE}"

TPCD_LOG_FILE=${TPCD_LOG}/query.${RUN_ID}_${i}.lo
g

TPCD_RPT_FILE=${TPCD_RPT}/query.${RUN_ID}_${i}.rp
t
  QRY_FILE=${QRY_DIR}/qtemp.${RUN_ID}_${i}.${SF}

########
# ${QPROG} tpcd/tpcd q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE}  \
#        > ${OUT}/qs.${RUN_ID}_${i}.${HID} 2>&1 &
########

  i=`expr $i + 1`
done

########
# (${SCRIPT_DIR}/audit_update_stream.sh
$START_SET_UPDATE $STOP_SET_UPDATE >
#${OUT}/us.${RUN_ID}_1.${HID} 2>&1 &)
########
# 1/23 for update
# sleep 10


wait
#svrmgrl <<!
#set echo on;
#connect tpcd/tpcd;
#set serveroutput on;
#execute dbms_alert.register('update_alert');
#execute dbms_alert.signal('my_alert','Queries
done');
#commit;
#declare
#        sss number;
#        mes varchar2(20);
#begin
#        dbms_alert.waitone('update_alert', mes,
sss);
#        dbms_output.put_line('Update done');
#end;
#/
#exit;
#!

# read < /tmp/th_pipe
# print > /tmp/th_pipe

M_EDATE=`date`
M_END=`$GTIME`
```

```
echo "Multi-stream ends at $M_END"
echo "Multi-stream ends at $M_END" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt

THR_INT=`echo $M_END - $M_START | bc`
echo "Throughput Measurement Interval is
$THR_INT" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "Throughput Measurement Interval is
$THR_INT"

echo ""
echo "-----------------------------------------
----"
echo ""
echo "Throughput Test completed at $M_EDATE" >>
${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
echo "Throughput Test completed at $M_EDATE"

THRUPUT=`expr 17 \* 3600 \* ${SF} \* 16`
THRUPUT=`echo "scale=2\n${THRUPUT}/${THR_INT}" |
bc`
echo "Final Throughput Metric is: $THRUPUT"
echo "Final Throughput Metric is: $THRUPUT" >> \
        ${TPCD_RPT}/tpcd.${RUN_ID}.${HID}.rpt
```

audit_uf1.sh

```
#!/bin/ksh
#================================================
==================+
#        Copyright (c) 1995  Oracle Corp, Redwood
Shores, CA       |
#                   OPEN SYSTEM PERFORMANCE GROUP
|
#                        All Rights Reserved
|
#================================================
==================+
# FILENAME
#    audit_uf1.sh
# DESCRIPTION
#    audit_uf1.sh
# USAGE
#    To execute UF1.
#================================================
==================

UPDATE_DIR=${TPCD_HOME}/UF_mstream
SCRIPT_DIR=${UPDATE_DIR}/scripts
LOG_DIR=${UPDATE_DIR}/log
GTIME=${SCRIPT_DIR}/gtime
PASSWD=tpcd/tpcd
LDR_PAR=5
LOAD_CTL=${TPCD_HOME}/loadctl/update

#
# Set SETNUM
#
if [ "$1" != "" ]; then
  SETNUM=$1
else
  if [ "$SETNUM" = "" ]; then
    SETNUM=1
  fi
fi

#
# Set PDDL/PDML degree
#
if [ "$PDML_DEGREE" = "" ]; then
  PDML_DEGREE=20
fi

if [ "$PDDL_DEGREE" = "" ]; then
  PDDL_DEGREE=10
fi


# first create the temp tables
load=1
if [ $load -eq 1 ]; then
  svrmgrl <<!
    connect $PASSWD

    alter session force parallel ddl parallel
(degree $PDDL_DEGREE);
    alter session force parallel dml parallel
(degree $PDML_DEGREE);
    alter session set
isolation_level=serializable;

    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          varchar2(10) ,
        l_linenumber        number,
        l_shipinstruct      varchar2(25) ,
        l_comment           varchar2(44)
    )
    parallel (degree $PDML_DEGREE instances 1)
    nologging;

    drop table temp_ord;
    create table temp_ord (
        o_orderdate         date ,
        o_orderkey          number,
        o_custkey           number,
        o_orderpriority     varchar2(15) ,
        o_shippriority      number ,
        o_clerk             varchar2(15) ,
        o_orderstatus       char(1) ,
        o_totalprice        number ,
        o_comment           varchar2(79)
    )
    parallel (degree $PDML_DEGREE instances 1)
    nologging;

    exit;
!

i=1
PID=""
START=`gtime`

#
# load into stagging area
```

```
#
echo "load start"
date
  while [ $i -le $LDR_PAR ]
  do

    sqlldr userid=$PASSWD
control=${LOAD_CTL}/tempitem.ctl \
      log=e:/tpcd814_work/log/hundgig/ti${i}.log
\

data=e:/tpcd814_work/update/data/hundgig/lineitem
.tbl.u${SETNUM}.${i} \
      direct=true parallel=true &

    sqlldr userid=$PASSWD
control=${LOAD_CTL}/tempord.ctl \
      log=e:/tpcd814_work/log/hundgig/to${i}.log
\

data=e:/tpcd814_work/update/data/hundgig/order.tb
l.u${SETNUM}.${i} \
      direct=true parallel=true &

   i=`expr $i + 1`

  done

  wait

fi

END_LOAD=`gtime`
echo "Load done. Elapsed Time is `echo $END_LOAD
- $START | bc`"

date

if [ "$LOAD_ONLY" = "yes" ]; then
  exit
fi

# now do the insert

svrmgrl <<!

  connect $PASSWD

  set timing on;
  set echo on

  alter session force parallel ddl parallel
(degree $PDDL_DEGREE);
  alter session force parallel dml parallel
(degree $PDML_DEGREE);
  alter session set isolation_level=serializable;

  analyze table temp_ord estimate statistics
sample 100 rows;
  analyze table temp_item estimate statistics
sample 100 rows;

  commit;

  insert /*+ parallel(orders, $PDML_DEGREE, 1) */
into orders
       (select * from temp_ord);
  insert /*+ parallel(lineitem, $PDML_DEGREE, 1)
*/ into lineitem
       (select * from temp_item);

  commit;

  exit;
!

wait

END=`gtime`

# Done

echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

audit_uf2.sh

```
#!/bin/ksh
#
#================================================
==================+
#        Copyright (c) 1995  Oracle Corp, Redwood
Shores, CA       |
#                   OPEN SYSTEM PERFORMANCE GROUP
|
#                         All Rights Reserved
|
#================================================
==================+
# FILENAME
#    uf2.sh
# DESCRIPTION
#    uf2.sh [-u <uid/passwd to login>] [-p
<program>] <run_id>
#              <scale factor> <pair number>
<parallelism>
# USAGE
#    To execute UF2.
#================================================
==================
UPDATE_DIR=${TPCD_HOME}/UF_mstream
SCRIPT_DIR=${UPDATE_DIR}/scripts
LOG_DIR=${UPDATE_DIR}/log
GTIME=${SCRIPT_DIR}/gtime
PASSWD=tpcd/tpcd
LDR_PAR=5
LOAD_CTL=${TPCD_HOME}/loadctl/update

#
# Set SETNUM
#
if [ "$1" != "" ]; then
  SETNUM=$1
else
  if [ "$SETNUM" = "" ]; then
    SETNUM=1
  fi
fi

#
# Set PDDL/PDML degree
#
if [ "$PDML_DEGREE" = "" ]; then
  PDML_DEGREE=20
fi
```

```
if [ "$PDDL_DEGREE" = "" ]; then
  PDDL_DEGREE=10
fi


# first create the temp tables
load=1
if [ $load -eq 1 ]; then
  svrmgrl <<!
    connect $PASSWD

    alter session force parallel ddl parallel
(degree $PDDL_DEGREE);
    alter session force parallel dml parallel
(degree $PDML_DEGREE);
    alter session set
isolation_level=serializable;

    drop table temp_okey;
    create table temp_okey (
       t_orderkey           number)
    parallel (degree $PDML_DEGREE instances 1)
    nologging;

    exit;
!

  i=1
  PID=""
  START=`gtime`

  #
  # load into stagging area
  #
  while [ $i -le $LDR_PAR ]
  do

    sqlldr userid=$PASSWD
control=${LOAD_CTL}/tempokey.ctl \
    log=e:/tpcd814_work/log/hundgig/ti${i}.log \

data=e:/tpcd814_work/update/data/hundgig/okeyuf2.
u${SETNUM}.${i} \
    direct=true parallel=true &

    i=`expr $i + 1`
  done

  wait

fi

END_LOAD=`gtime`
echo "Load done. Elapsed Time is `echo $END_LOAD
- $START | bc`"


if [ "$LOAD_ONLY" = "yes" ]; then
  exit
fi

svrmgrl <<!

  connect $PASSWD ;

  set timing on;
  set echo on

  alter session force parallel ddl parallel
(degree $PDDL_DEGREE);
  alter session force parallel dml parallel
(degree $PDML_DEGREE);
  alter session set isolation_level=serializable;

  commit;

  delete /*+ ordered use_nl(orders) index(orders,
o_okey) */ orders
  where o_orderkey in (select /*+ nl_sj */ * from
temp_okey);

  delete /*+ ordered use_nl(lineitem)
index(lineitem, l_ored) */ lineitem
  where l_orderkey in (select /*+ nl_sj */ * from
temp_okey);

  commit;

 exit;
!

END=`gtime`

# Done

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

Appendix F. Database Layout

DB Entry	Volumes	Disks/Volume
lineitem	ts_l_100g[1-3]	24
orders	ts_o_100g[1-3]	24
partsupp	ts_ps_100g[1-3]	24
parts	ts_p_100g[1-3]	24
supplier	ts_s_100g[1-3]	24
customer	ts_c_100g[1-3]	24
Materialized	ts_ma_lineitem_100g[1-3]	24
Views	ts_mj_locs_100g[1-6]	24
	ts_mllog_100g[1-3]	24
	ts_maint_100g[1-3]	24
	ts_mj_ppssrn_100g[1-3]	24
Indexes	ts_ij_l_sdate_100g[1-3]	24
	ts_ij_l_rdate_100g[1-3]	24
	ts_ij_l_odate_100g[1-3]	24
	ts_ij_l_pkey_100g[1-3]	24
	ts_ind_lored_100g[1-3]	24
	ts_ind_oclokd_100g[1-3]	24
	ts_ij_lrid_100g[1-3]	24
	ts_ij_orid_100g[1-3]	24
	ts_ij_nullrid_100g[1-3]	24
	rs_ind_okey_100g[1-3]	24
	ts_ij_ppsn_100g[1-3]	24
	ts_ij_nnpsps_100g[1-3]	24
	ts_p_tp_100g[1-3]	24
	ts_p_cbp_100g[1-3]	24
Constraint	ts_constr_100g[1-3]_	24
Temp	ts_temp_100g[1-15]	24
System (contains nation and region)	sys_100g[1-3]	24
Undo	undo_100g[1-3]	24
Log files	log_100g[1-2]	2

Volume for datafiles was configured as 5 stripe sets of 24 disks each, for a total of 120 disks.
It containing the indexes, materialized views, tables, rollback segments, system file, and temp file.

Volume for log files was configured as a mirrored set of 2 disks.

Appendix G.
Initial Ten Rows of Tables

10rows.sql

```
rem This gets 10 rows from each table and spools
it to
rem the file 10rows.out . The output is required
for
rem appendix F of the FDR

spool out/10rows.out
select * from orders where rownum < 11;
select * from lineitem where rownum < 11;
select * from customer where rownum < 11;
select * from parts where rownum < 11;
select * from supplier where rownum < 11;
select * from partsupp where rownum < 11;
select * from nation where rownum < 11;
select * from region where rownum < 11;
exit
```

10rows.out

```
O_ORDERDA O_ORDERKEY  O_CUSTKEY O_ORDERPRIORITY
O_SHIPPRIORITY O_CLERK         O
--------- ---------- ---------- --------------- -
------------ --------------- -
O_TOTALPRICE
------------
O_COMMENT
-----------------------------------------------
------------------------------
11-APR-95   37606500    9946504 2-HIGH
0 Clerk#000058765 P
    129932.6
5A i354142zCNO
RwzilSmg5PkNgSl21hCPzAiO1j7Rl4415zAk360Q1kB

13-FEB-93   37607428    1499101 1-URGENT
0 Clerk#000057964 F
     3955.95
6nA4z1y5g1S0 knClQQ5lQwnlM3
knynznCMg3yiQjhkN6x1Cx1ySS31

O_ORDERDA O_ORDERKEY  O_CUSTKEY O_ORDERPRIORITY
O_SHIPPRIORITY O_CLERK         O
--------- ---------- ---------- --------------- -
------------ --------------- -
O_TOTALPRICE
------------
O_COMMENT
-----------------------------------------------
------------------------------

29-MAY-92   37607655    2204441 3-MEDIUM
0 Clerk#000014656 F
   140509.18
 kilyyBwn46yOn7Czj32Oh1Qm

09-JAN-92   37608293     609958 2-HIGH
0 Clerk#000054140 F
    81959.89

O_ORDERDA O_ORDERKEY  O_CUSTKEY O_ORDERPRIORITY
O_SHIPPRIORITY O_CLERK         O
--------- ---------- ---------- --------------- -
------------ --------------- -
O_TOTALPRICE
------------
O_COMMENT
-----------------------------------------------
------------------------------
 662QP6gm7B5znhNjmlg6mgzC3whh0gR50h6L0Q7

08-JUN-95   37608643    9544498 1-URGENT
0 Clerk#000077964 O
     15542.8
BA05AOwjOnBwnRNSgw4MBMwxRzjNR7
m25Mmm0OSi42hwj7S3C05wk0m5QwwQ0zA21i

08-AUG-92   37608804   10195216 4-NOT SPECIFIED
0 Clerk#000061756 F

O_ORDERDA O_ORDERKEY  O_CUSTKEY O_ORDERPRIORITY
O_SHIPPRIORITY O_CLERK         O
--------- ---------- ---------- --------------- -
------------ --------------- -
O_TOTALPRICE
------------
O_COMMENT
-----------------------------------------------
------------------------------
    207886.5
7C4ww4CM7ng0Q5wyQ17wRSwwzm3w5gP5wQ1AgynRnNNOSM

25-OCT-92   37609059    5950237 1-URGENT
0 Clerk#000042930 F
   197212.37
0NhP6SAQ23j360jQi0QQC4xl3jj3Rwzi NOSQ5B5Q6


O_ORDERDA O_ORDERKEY  O_CUSTKEY O_ORDERPRIORITY
O_SHIPPRIORITY O_CLERK         O
--------- ---------- ---------- --------------- -
------------ --------------- -
O_TOTALPRICE
------------
O_COMMENT
-----------------------------------------------
------------------------------
01-JAN-96   37609123   11170144 2-HIGH
0 Clerk#000037776 O
   151529.49
LSQ34AwyL4R727k7m 4hgwwkO7QnMk0MNl231lnmzi

17-SEP-95   37609479    1635316 4-NOT SPECIFIED
0 Clerk#000017308 O
   190432.47
CxSglnx6Om4Sjm63NPi5L6SiS5B5N7iP6jN0Pzg5NOn5L4

O_ORDERDA O_ORDERKEY  O_CUSTKEY O_ORDERPRIORITY
O_SHIPPRIORITY O_CLERK         O
--------- ---------- ---------- --------------- -
------------ --------------- -
O_TOTALPRICE
------------
O_COMMENT
-----------------------------------------------
------------------------------

13-MAY-98   37609894   13329031 3-MEDIUM
```

```
0 Clerk#000025752 O
   297233.81
6
yMPB1nMjg4g2lNNzRi662yy22LxA5yNyj1MN45ARSgBwBNkhj
3P nCk2


10 rows selected.


L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE
L_SUPPKEY L_QUANTITY L
--------- ---------- ---------- --------------- -
--------- ---------- -
 L_PARTKEY L      L_TAX L_COMMITD L_RECEIPT
L_SHIPMODE L_LINENUMBER
--------- - ---------- --------- --------- -----
----- ------------
L_SHIPINSTRUCT            L_COMMENT
------------------------- ------------------------
---------------------
26-FEB-98  339983619          0         39242.8
318013         20 N
  17067995 O          0 30-JAN-98 28-FEB-98 REG
AIR               2
DELIVER IN PERSON         yk3R6LPg3l1h3  Ly16
Rjn50CO4

22-FEB-98  339983619         .1        80882.49
279997         39 N
   1779994 O        .06 12-JAN-98 05-MAR-98 RAIL
3
COLLECT COD               mmly2MmwyCxlg4
3hPR6myPLABRRx5wN6yn3jx5

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE
L_SUPPKEY L_QUANTITY L
--------- ---------- ---------- --------------- -
--------- ---------- -
 L_PARTKEY L      L_TAX L_COMMITD L_RECEIPT
L_SHIPMODE L_LINENUMBER
--------- - ---------- --------- --------- -----
----- ------------
L_SHIPINSTRUCT            L_COMMENT
------------------------- ------------------------
---------------------

17-FEB-98  339984294        .01         30357.8
276466         22 N
  11026454 O        .07 26-FEB-98 20-FEB-98 AIR
2
NONE                      PRPCNw3km6O3LxBgy
02yliCMk3i

22-FEB-98  339984327        .08         9235.04
965292          8 N
  17215240 O        .03 02-MAR-98 15-MAR-98 RAIL
3

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE
L_SUPPKEY L_QUANTITY L
--------- ---------- ---------- --------------- -
--------- ---------- -
 L_PARTKEY L      L_TAX L_COMMITD L_RECEIPT
L_SHIPMODE L_LINENUMBER
--------- - ---------- --------- --------- -----
----- ------------
L_SHIPINSTRUCT            L_COMMENT
------------------------- ------------------------
---------------------
DELIVER IN PERSON
yBwP1hA3mM3y4wnLzwRj7Qz5lmzylkR17lhi

20-FEB-98  339984515         .1        92082.24
190888         48 N
   9940878 O          0 06-APR-98 28-FEB-98 FOB
3
NONE                      yhSzkCjPQzj

25-FEB-98  339984517        .06        41761.33
419853         23 N

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE
L_SUPPKEY L_QUANTITY L
--------- ---------- ---------- --------------- -
--------- ---------- -
 L_PARTKEY L      L_TAX L_COMMITD L_RECEIPT
L_SHIPMODE L_LINENUMBER
--------- - ---------- --------- --------- -----
----- ------------
L_SHIPINSTRUCT            L_COMMENT
------------------------- ------------------------
---------------------
   2669846 O        .06 31-DEC-97 27-FEB-98 SHIP
2
NONE                      igyO SSww6h3mM47
mzgP0jCkxw40N6ix0MC34hwwN

17-FEB-98  339984517        .07        44069.76
338861         24 N
  12088848 O        .01 21-JAN-98 08-MAR-98 RAIL
3
DELIVER IN PERSON         AlxMzNiR2n2x


L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE
L_SUPPKEY L_QUANTITY L
--------- ---------- ---------- --------------- -
--------- ---------- -
 L_PARTKEY L      L_TAX L_COMMITD L_RECEIPT
L_SHIPMODE L_LINENUMBER
--------- - ---------- --------- --------- -----
----- ------------
L_SHIPINSTRUCT            L_COMMENT
------------------------- ------------------------
---------------------
23-FEB-98  339985029        .06         64873.6
613412         44 N
    363411 O        .04 19-FEB-98 11-MAR-98 MAIL
2
COLLECT COD
gMRBwxlkiPNmMOzOCy7x10h2k6Q64AMg

22-FEB-98  339985089        .02         34345.5
560576         21 N
   1560575 O        .08 12-MAR-98 03-MAR-98 AIR
3
NONE
ijPNyw16B7540hlLNzkMRMx361yhP00

L_SHIPDAT L_ORDERKEY L_DISCOUNT L_EXTENDEDPRICE
L_SUPPKEY L_QUANTITY L
--------- ---------- ---------- --------------- -
--------- ---------- -
 L_PARTKEY L      L_TAX L_COMMITD L_RECEIPT
L_SHIPMODE L_LINENUMBER
--------- - ---------- --------- --------- -----
----- ------------
```

```
L_SHIPINSTRUCT            L_COMMENT
------------------------- ----------------------
---------------------

18-FEB-98  339985095        .06        13249.34
966089         13 N
  17466054 O        .05 27-JAN-98 11-MAR-98 FOB
3
DELIVER IN PERSON
hxMyN0xkNRzlM2hzSlLPxRzzgCkjQ


10 rows selected.


 C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME
---------- ---------- ------------ ---------------
----------
C_ADDRESS                                 C_PHONE
C_ACCTBAL
---------------------------------------- --------
------- ----------
C_COMMENT
---------------------------------------------------
--------------------------------
   1886497 AUTOMOBILE          17
Customer#001886497
S3iQ4hk27ixB377hMk363yN3jxA1302PS2wy     27-428-
244-6066    9718.72
3LBkyn O4RQPBzwnO52C4yxBzzjyQC1jNOg433zA0m6y QB

   1886500 AUTOMOBILE          14
Customer#001886500
n6N3xwS5CRnBCy 3C                         24-335-
101-3204     5763.5
OlnL345hRN2PBS0g0wxS52Axk
7iMj5hn2kSlM4Mh5hBxiQxAx

 C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME
---------- ---------- ------------ ---------------
----------
C_ADDRESS                                 C_PHONE
C_ACCTBAL
---------------------------------------- --------
------- ----------
C_COMMENT
---------------------------------------------------
--------------------------------


   1886504 AUTOMOBILE           6
Customer#001886504
14RM7Rzk N40  16Am3yx2PPS34OB6lSk4 5kgh  16-569-
639-4600    5389.51
Nh0LzNCP5x31LQSOjByi7mLCkyCMx1Rx2A

   1886506 AUTOMOBILE          16
Customer#001886506
PlkOnmN1gOQRmBRBM2iP 31LB PBlL4P3O7R764  26-588-
390-6510    8684.99

 C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME
---------- ---------- ------------ ---------------
----------
C_ADDRESS                                 C_PHONE
C_ACCTBAL
---------------------------------------- --------
------- ----------
C_COMMENT
---------------------------------------------------
------------------------------
hAQy222 izLlB55wMzLB1kPSNg Bzwgi6RizgNh2Qi
iixLm7zPzg2mjjx2zy

   1886510 AUTOMOBILE           4
Customer#001886510
y6A5wOh2wCAyCR5gAOnzR6LyQmByli6          14-444-
629-9093    4460.47
2AAPN1ilL4QS4ihi1OmnB21h R45lh

   1886521 AUTOMOBILE          11
Customer#001886521

 C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME
---------- ---------- ------------ ---------------
----------
C_ADDRESS                                 C_PHONE
C_ACCTBAL
---------------------------------------- --------
------- ----------
C_COMMENT
---------------------------------------------------
--------------------------------
7CP2lh1lkBg                               21-886-
805-1169    2755.63
yBQ14PkAPOR2x346Q0ggA53wR3Mz2izxPzOQ1Bzh0wPAOP6SQ
55

   1886530 AUTOMOBILE          20
Customer#001886530
1PM5BL5ikCS7SO RhBgC217xgng              30-564-
423-3316    3569.23
CQn6MNClm0RQQPSgP4C5gBzmMA2nPOSMl3kPMyyiiO jAmNRl
2PgSkSg3MM1wjxQy6ixyy36N2OwP m
M 77A7hPRwOlL

 C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME
---------- ---------- ------------ ---------------
----------
C_ADDRESS                                 C_PHONE
C_ACCTBAL
---------------------------------------- --------
------- ----------
C_COMMENT
---------------------------------------------------
--------------------------------


   1886532 AUTOMOBILE          12
Customer#001886532
i1g2SQgRxQ4O                              22-835-
167-4819    6470.67
yM7hC
h2By4NkQjAiRhMNhxQjkLmn41n0h121B3CwNRPw0zAwg QC

   1886535 AUTOMOBILE          17
Customer#001886535
N0xz3Aj0CPj70n0kkNmgigR5                  27-727-
810-2796    8145.82

 C_CUSTKEY C_MKTSEGME C_NATIONKEY C_NAME
---------- ---------- ------------ ---------------
----------
C_ADDRESS                                 C_PHONE
C_ACCTBAL
---------------------------------------- --------
------- ----------
C_COMMENT
---------------------------------------------------
--------------------------------
```

```
LzlBMiBQkzPmnSx0PB67i 0O4OR3OC hyBCMS6

   1886545 AUTOMOBILE          11
Customer#001886545
hMy5mg1z3zyiN2MOP0Aimh6P                 21-397-
694-1261    2866.86
NSB04il6Rkzmw6
n0L6RQ0hRSgmlOhyQPn5kQyx4jnwgjnkjgP


10 rows selected.


 P_PARTKEY P_TYPE                          P_SIZE
P_BRAND
---------- ------------------------- ---------- -
---------
P_NAME
P_CONTAINE
-------------------------------------------------
------ ----------
P_MFGR                    P_RETAILPRICE P_COMMENT
------------------------- ------------- ---------
--------------
   1727028 LARGE BRUSHED TIN                   1
Brand#52
navajo royal turquoise blush grey
MED BOX
Manufacturer#5                   1054.94
x4x6w6QkS74P

   1727041 PROMO ANODIZED TIN                  1
Brand#55
saddle navy green metalic thistle
JUMBO JAR
Manufacturer#5                   1067.96
2jmymQAjB3h5My0LyN7

 P_PARTKEY P_TYPE                          P_SIZE
P_BRAND
---------- ------------------------- ---------- -
---------
P_NAME
P_CONTAINE
-------------------------------------------------
------ ----------
P_MFGR                    P_RETAILPRICE P_COMMENT
------------------------- ------------- ---------
--------------

   1727084 LARGE BRUSHED COPPER                1
Brand#52
lavender spring coral puff cornflower
LG CASE
Manufacturer#5                     1111
SxjO7m42z3wRRwQnQkQ

   1727124 ECONOMY BRUSHED STEEL               1
Brand#22
royal ghost orchid black beige
SM PACK

 P_PARTKEY P_TYPE                          P_SIZE
P_BRAND
---------- ------------------------- ---------- -
---------
P_NAME
P_CONTAINE
-------------------------------------------------
------ ----------
P_MFGR                    P_RETAILPRICE P_COMMENT
------------------------- ------------- ---------
--------------
Manufacturer#2                   1151.04 2l32N

   1727130 ECONOMY PLATED BRASS                1
Brand#15
light misty ghost burnished burlywood
JUMBO PKG
Manufacturer#1                   1157.05
jO4lLkiCQ7

   1727146 MEDIUM BRUSHED BRASS                1
Brand#42

 P_PARTKEY P_TYPE                          P_SIZE
P_BRAND
---------- ------------------------- ---------- -
---------
P_NAME
P_CONTAINE
-------------------------------------------------
------ ----------
P_MFGR                    P_RETAILPRICE P_COMMENT
------------------------- ------------- ---------
--------------
papaya yellow olive deep tomato
SM BOX
Manufacturer#4                   1173.06 Pzj13h0g
h6m

   1727158 SMALL ANODIZED COPPER               1
Brand#41
sky metalic linen wheat peach
LG JAR
Manufacturer#4                   1185.07 6
xMBnhgOL5224Nm2k7y


 P_PARTKEY P_TYPE                          P_SIZE
P_BRAND
---------- ------------------------- ---------- -
---------
P_NAME
P_CONTAINE
-------------------------------------------------
------ ----------
P_MFGR                    P_RETAILPRICE P_COMMENT
------------------------- ------------- ---------
--------------
   1727189 ECONOMY BURNISHED NICKEL            1
Brand#31
white chiffon midnight tan blue
WRAP JAR
Manufacturer#3                    1216.1 MgMgw00CL

   1727249 MEDIUM ANODIZED STEEL               1
Brand#25
green blush cornsilk spring honeydew
JUMBO JAR
Manufacturer#2                   1276.16 OA5S6

 P_PARTKEY P_TYPE                          P_SIZE
P_BRAND
---------- ------------------------- ---------- -
---------
P_NAME
P_CONTAINE
```

```
------------------------------------------------
------ ----------
P_MFGR                    P_RETAILPRICE P_COMMENT
------------------------- ------------- ---------
--------------

   1727289 ECONOMY POLISHED STEEL             1
Brand#25
midnight navy turquoise magenta cream
JUMBO BAG
Manufacturer#2                    1316.2
BwQxB2kyzPL


10 rows selected.


 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------
    199527          12
44jmkm6Al4gORi iOR513OS7SC0135gh6zA47h3 zzzSRnm
7g22ROwP7M7L M53xSyNhny3AR6RARxP
xw0wjOgA6Owm2MB
Supplier#000199527
P0AkSMNgMlMjS1xLB7klOxzN
22-678-210-3075    4994.67

 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199528          14
mj7g6CgCAmOxSwCz7nClOm3Rx6i 1jOhSSLBhi2
5mnzA5lB7yw31zMR 10Pilwi3BN3m1SiL
Supplier#000199528        OQRxAMC71m
j5hQwByQMP5nnMOPRwyBASRn
24-800-485-5960    2437.54

 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199529          18
7RPO
2B2kjzO4mP0hN7hiwzmPyw5C717PLxRhlP6R3y5Cwgy1PPnO
Supplier#000199529        Bxwh56i12iC
28-215-797-2862    1644.58

 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199530          9
i1QlnO4B OBjASC2PMi3gM7LLB3Q0iOj
Supplier#000199530        whjNijm7gAi2O7
19-467-860-2989     6486.5

 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199531          14
RligkSNiQxx3mLjNAiLP
jwyAQh6O05l5nzAmShO052n5AS5g47jM5jRwB2B6M5MQzR
Supplier#000199531        CxgylnzS6A6LPjP
24-244-750-9676    6658.31

 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199532          4
R1QgOMO53xw1kn07Qx y2NhlPygkR5Rmljw0006y6Plyl
MNO27h205nMBxyz2wACwOLhmA3g0
Supplier#000199532
wRCNz4hB5h23wP12gLL76nR615hSy6mSMQ
14-702-565-5215    1220.57

 S_SUPPKEY S_NATIONKEY
---------- ------------
S_COMMENT
------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199533          8
14O540Lx
6L3mPyxRM4lOL20i00Pl1lSmwwnLOk1jxQgg4w7LykO4SyQlR
26iihAmQhQm2g7j1NNi2Sj
Supplier#000199533        hl0OxPQ7hNx64yBigS3P
18-867-310-1749     8138.6
```

```
 S_SUPPKEY S_NATIONKEY
---------- -----------
S_COMMENT
---------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199534           5
4BQ3jnlj ANx3hgyMMy72 4QNNn
Supplier#000199534         5iQxmL2m07Cg3RkP
ShgOCOCiNgCRhSh2B5xNRM
15-451-563-7139    9466.31

 S_SUPPKEY S_NATIONKEY
---------- -----------
S_COMMENT
---------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------

    199535          22
4BMRNnRnO63y0P1n7mgM1gk413O5LyQkn237A1h3z17z42lin
B3k5CAxBgkC1Si4BzN  B03 B wM4Mz
Q1lQhAP
Supplier#000199535        iPn0m72S7PSgQlMiN0
NSyySCgyOmLm0

 S_SUPPKEY S_NATIONKEY
---------- -----------
S_COMMENT
---------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------
32-630-557-3343    5416.87

    199536           3
kwljyP Lxhm 6PLORB1L4nw3RCLj0jBQLM3Nnx5ihC1
Supplier#000199536
L4z0yRl7kz3nnkCkCnz5h2h63z

 S_SUPPKEY S_NATIONKEY
---------- -----------
S_COMMENT
---------------------------------------------------
--------------------------------
S_NAME                    S_ADDRESS
------------------------- ------------------------
-----------------
S_PHONE          S_ACCTBAL
--------------- ----------
13-202-915-8939    3306.46


10 rows selected.
```

```
PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY
---------- ---------- ------------- -----------
PS_COMMENT
---------------------------------------------------
--------------------------------
  17590194     590195         73.99        6757
l3LkAnhBS 7 LgAzwCi1nn2N5 BLSCS3
2N1M1Mk3likAh5yC3BMB0L7627C5Cw 0L13l1nPRN3Ll h5
MMwjxBLmA3lkCLnn1Q5gRxxP3ALP03S4g4j

  17590194     840212         29.37        2675
jP4LySQ01SRglAzBk5khOShR1S224BC1ly5NA2gC4055OPS75
O3wLmR2y6j2k515Pkz4QNLCOR6hRP6g
 MwRhRSkh264hL0l7x1LS x4P

  17590194      90229        454.86        6719

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY
---------- ---------- ------------- -----------
PS_COMMENT
---------------------------------------------------
--------------------------------
4 Slz65QN31kgCN S 2PQRC014kgLzlm7izgSAPSSM7LPhQ1
Rj03i010yjOSMi66CzxAm3Pww560Sxi
47RNMBmLnnxz7xQk3jw1SAxz0Bh6gSM5nghOz

  17590194     340246        704.63        3241
5SB0gM xl7kQihOnN32O6
i170P0ygLSzk1mBO44AA3wz5l06S2NjhSSx
N7L474S2kBiM3Ol33m 3y7
Oji7AB7MByiwLx31wxQByOi1n57C22MkQNhl

  17590195     590196        700.94        6637
yj1ARyy ixA1Oi4CmCzNMP4101Nj7wnwjkn
hzC2j4LlMO3AySBRmCN

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY
---------- ---------- ------------- -----------
PS_COMMENT
---------------------------------------------------
--------------------------------

  17590195     840213        609.64        6736
zj6OO3w4mLS0kA1ALm3Bi4 Ng5ChR3mjx2yNQj h50Bm6M
O2LhnL6B4B3S kQ m4Qh3giCkm5AOg1SA
 gnx SAN3Bij7PAxBnx5gMSPQR0hCACzMkn
QgOQhMkzmN3Ozm2k3RiiMN0MO3xk7zRS3Rn2zhw4i5mR
x42

  17590195      90230        703.55         474
R h1zBiNjMhwj1h0Alm31g4Qm nm0Sw1j0j433AORA5
6yOm6giA0l33BM CAAkkyO7AL


PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY
---------- ---------- ------------- -----------
PS_COMMENT
---------------------------------------------------
--------------------------------
  17590195     340247        496.72        5538
Sy1BjyLOAPl6zkyMML5NLjNAPjPOnCjM0x2SRPB4zzlgh
h1NBLxy 6k3 k673wSROggMlPSClB01ChA
jQ h42jhQPCgNN1gSg5LNNkP3Qx

  17590196     590197        847.29         985
w54Oj41N6PLijMl46Oy6kOgMk3BC423nzi0Q6ALwA56B3MO5m
nlSN43gwP1wjQgLjPA7y0216gAlS6hR
gwgSnQ7P31BRxxxLOMm5P10jxAPO 4MByhP
y243mk65w6hlkP
```

```
  17590196     840214        766.66        1371

PS_PARTKEY PS_SUPPKEY PS_SUPPLYCOST PS_AVAILQTY
---------- ---------- ------------- -----------
PS_COMMENT
------------------------------------------------
--------------------------------
jARnxCj7wSNRMPOggRB26hMiwkNxl2Pzh0Mnj
L4m5A27kkQ5n6B6AM21LSOAwhOyN5yOO M6miQ7AiM
g3Rwk7CxOQm7gwB n3R05nQBCAh2lwgj2m


10 rows selected.


N_NATIONKEY N_NAME                      N_REGIONKEY
----------- ------------------------- -----------
N_COMMENT
------------------------------------------------
--------------------------------
          0 ALGERIA                             0
2Cxhl7 L1iwk6hMh300izngN32CPwCikyLk6khMzSRA

          1 ARGENTINA                           1
zQn3Okwz1wLn7PLS3OhCgn56kP5PyRikgi1B7lL

          2 BRAZIL                              1
gLmS0nACAmnBCj2klki7RCPNgPxnCOjNg4k
OiAg57COSOm1NwCnOyLx40R SC y20gPPAkNk5hxRhR5
OmgS1iPQQzNAxPL30n67OgyC l617Sh4LS

N_NATIONKEY N_NAME                      N_REGIONKEY
----------- ------------------------- -----------
N_COMMENT
------------------------------------------------
--------------------------------

          3 CANADA                              1
 4yMO AhnQ5Lh wzQAM662Aw1ByCl7CxmzRwNR5nAlO4 x

          4 EGYPT                               4
l1im5126  Cxj
NMQmLxOikni02j2m3Ah4yNR1QQiL507j2QSlyN

          5 ETHIOPIA                            0
NS7n LSOP Oz5n1AlB2S02nN0lMh4SBxP iRhBO 047R26
2BlM

N_NATIONKEY N_NAME                      N_REGIONKEY
----------- ------------------------- -----------
N_COMMENT
------------------------------------------------
--------------------------------

          6 FRANCE                              3
3mjmizl S 3L3k2hNNhNlP4w370xRxyN15wn

          7 GERMANY                             3
z nOP4RkwO CmzBB 516mAg lByw4OM3QyNPA

          8 INDIA                               2
MNlR5RCiRMj1l11wjN7Myn M1lylNlMmBQl7PL4C
kkxQkgPQ7i3w6B67R2QkOO40xl4Q2iw76jRL7i

N_NATIONKEY N_NAME                      N_REGIONKEY
----------- ------------------------- -----------
N_COMMENT
------------------------------------------------
--------------------------------
lhR5Q 0xC7RRm5iQ2NAx2LiBm3QiO27j

          9 INDONESIA                           2
SjPmQO71Lj
7ABj6MxlAQk3nLwi73BPxzCwjzMn4zlLzgg6nnz0j0w
zxC66gP6ykRPMg


10 rows selected.


R_REGIONKEY R_NAME
----------- -------------------------
R_COMMENT
------------------------------------------------
--------------------------------
          0 AFRICA
xSx31zz31Cl1z4OAnmm05AjiOxC3AMMNOgC0kACgwngg3glP7
LLLywlQy7R

          1 AMERICA
kgyh3LSnC72k6zlAz0LP3k2L4QB1QL1O673OjO1SPj0ngQ7CO
100SBgmgRQ4lgPCMk21A425iklyAR4y
BRAwR4Cm5miNw 4jl13mMnxw17B

          2 ASIA
NSg6xlMlA1lzm6mOR0Ajx nhRA77NgRxBwL1M6Py
RjySB3RLwkyPkwMM2R1BQ xAzkOgkjmll0gAghi

R_REGIONKEY R_NAME
----------- -------------------------
R_COMMENT
------------------------------------------------
--------------------------------
nP5inmNmR76MlijMS3S2zxONR15

          3 EUROPE
zlSL7Qwg12hMBL5lhlz0M45QkjShwSyiO04MLOh7wn1ARLQPy
PAyAiil57611Li7AlnR1S RQ4SLny7B
2Ryj5P66MLhn NxhwB4C3ig0SO

          4 MIDDLE EAST
RllxmhPLz3Cy2mNlg4QMBnNASM ACki MPki7Oi
```

Appendix H.
Pricing:
Reseller Price Quotes

Visual C++ price: obtained on web catalog, SOS developers:

www.sosdevelopers.com

MICROSOFT VISUAL C++ PRO STD CD ROM 3690 FHT

3690/6.55957= 563 Euros

ORACLE

February 10, 1999

Mr. Jean-Francois.Lemerre
BULL, B.P. 208,
38432 Echirolles
Cedex,
France

Dear Mr. Lemerre,

In response to your inquiry, here is the pricing inEuro's and availability information for the specified configuration:

Oracle8i License:	166,531
Oracle Silver support for five years:	221,244
Oracle8i version 8.1.5.1 availability date:	May 18, 1999

Sincerely,

Ray Glasstone
Oracle Corp.

MKS France

11 C, Quai Conti, 78430 LOUVECIENNES - FRANCE
Tél. (33) 01 30 82 27 62 Fax. (33) 01 30 82 72 78 E-Mail mks@scientific.fr

DESTINATAIRE	:	BULL
NUMERO DE FAX	:	04 76 29 75 18
A L'ATTENTION DE	:	M. Jean François LEMERRE
EMETTEUR	:	Mr Claude LEVY
REFERENCE	:	M.13.99.786
OBJET	:	MKS Toolkit

NOMBRE DE PAGES (FEUILLE DE GARDE INCLUSE)	:	1
DATE DE TRANSMISSION	:	12 Février 1999

Le prix de vente d'une licence MKS TOOLKIT est de 507,66 Euro H.T.

Salutations distinguées.

C. LEVY