HP Integrity Superdome Enterprise Server
using
HP-UX 11.i V2 64-bit
and
Oracle 10g Database Enterprise Edition with Partitioning

TPC Benchmark™ H
Full Disclosure Report

Second Edition

December 31, 2003

Second Edition - December 31, 2003

Printed in U.S.A., December 31, 2003.

Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on the HP Integrity Superdome Enterprise Server, in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 2.1.0. The operating system used for the benchmark was HP-UX 11.i V2 64-bit; the DBMS was Oracle 10g.

Standard and Executive Summary Statements

The pages following this preface contain the Executive Summary and Numerical Quantities Summary of the benchmark results.

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results and the pricing model used to calculate the cost per QphH was audited by Francois Raab, InfoSizing, to verify compliance with the relevant TPC specifications.

TPC Benchmark H Overview

The TPC Benchmark ™ H (TPC-H) is a decision support benchmark. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates decision support systems that

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

TPC-H evaluates the performance of various decision support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries(e.g., via a point and click GUI interface);
- 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.

The TPC-H operations are modeled as follows:

- The database is continuously available 24 hours a day, 7 days a week, for ad-hoc queries from multiple end users and updates against all tables, except possibly during infrequent (e.g., once a month) maintenance sessions;
- The TPC-H database tracks, possibly with some delay, the state of the OLTP database through on-going updates which batch together a number of modifications impacting some part of the decision support database;
- Due to the world-wide nature of the business data stored in the TPC-H database, the queries and the updates may be executed against the database at any time, especially in relation to each other. In addition, this mix of queries and updates is subject to specific ACIDity requirements, since queries and updates may execute concurrently;

- To achieve the optimal compromise between performance and operational requirements the database administrator can set, once and for all, the locking levels and the concurrent scheduling rules for queries and updates.

The minimum database required to run the benchmark holds business data from 10,000 suppliers. It contains almost ten million rows representing a raw storage capacity of about 1 GB. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g. 3000 GB), as defined in Clause 4.1.3.

The performance metrics reported by TPC-H measure multiple aspects of the capability of the system to process queries. The TPC-H metric at the selected size (QphH@Size) is the performance metric. To be compliant with the TPC-H standard, all references to TPC-H results for a given configuration must include all required reporting components (see Clause 5.4.7). The TPC believes that comparisons of TPC-H results measured against different database sizes are misleading and discourages such comparisons.

The TPC-H database must be implemented using a commercially available database management system (DBMS), and the queries executed via an interface using dynamic SQL. The specification provides for variants of SQL, as implementers are not required to have implemented a specific SQL standard in full. TPC-D uses terminology and metrics that are similar to other benchmarks, originated by the TPC and others. Such similarity in terminology does not in any way imply that TPC-H results are comparable to other benchmarks. The only benchmark results comparable to TPC-H are other TPC-H results compliant with the same revision.

Despite the fact that this benchmark offers a rich environment representative of many decision support systems, this benchmark does not reflect the entire range of decision support requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-H approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-H should not be used as a substitute for a specific customer application benchmarking when critical capacity planning and/or product evaluation decisions are contemplated.

Benchmark sponsors are permitted several possible system designs, provided that they adhere to the model described in Clause 6. A full disclosure report (FDR) of the implementation details, as specified in Clause 8, must be made available along with the reported results.

General Implementation Guidelines

The purpose of TPC benchmarks is to provide relevant, objective performance data to industry users. To achieve that purpose, TPC benchmark specifications require that benchmark tests be implemented with systems, products, technologies and pricing that:

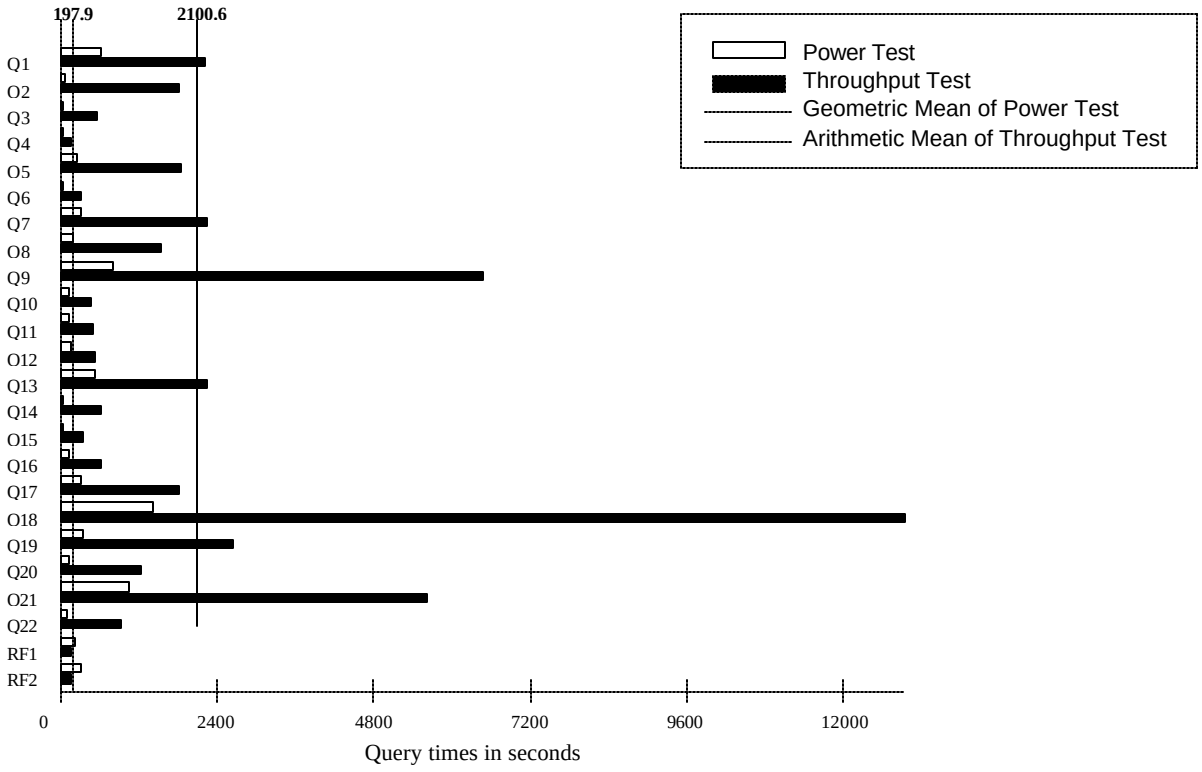
- Are generally available to users;
- Are relevant to the market segment that the individual TPC benchmark models or represents (e.g. TPC-H models and represents complex, high data volume, decision support environments);
- Would plausibly be implemented by a significant number of users in the market segment the benchmark models or represents.

Hewlett-Packard Company does not warrant or represent that a user can or will achieve performance similar to the benchmark results contained in this report. No warranty of system performance or price/performance is expressed or implied by this report

invent ORACLE	**HP Integrity Superdome Enterprise Server**	TPC-H Rev 2.1.0 Report Date: December 31, 2003

Total System Cost	Composite Query per Hour Metric	Price/Performance
$4,922,070	**45,247.8** **QphH@3000GB**	**$109** **QphH@3000GB**

Database Size	Database Manager	Operating System	Other Software	Availability Date
3000 GB*	**Oracle 10g Database Enterprise Edition with Partitioning**	**HP-UX 11.i V2 64-bit**	**None**	**March 25, 2004**



Database Load Time = 05:42	Load Includes Backup: N	Total Data Storage/Database Size = 26.33	
RAID (Base Tables Only): N	RAID (Base Tables and Auxiliary Data Structures): N		RAID (All): Y

System Configuration

Processors:	64 Itanium2 1.5GHz, 6MB L3 Cache
Memory:	256 GB
Disk Drives:	1 hp surestore disk system 2100 with 4 - 36GB 1-36GB LP 10K LVD SE U320 HDD disks and 72 SureStore VA7110 (with a total of 1080 73GB 15K RPM LVD disks)
Total Disk Storage	78984GB (In this calculation one GB is defined as 1024*1024*1024 bytes)
Lan Controllers	1 HP 10/100 Base-T Lan Adapter

*Database Size includes only raw data (e.g. no temp, index, redundant storage space, etc.)

HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date: December 31, 2003

Description	Part Number	Source	Reference Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
Super Dome left chassis	A5201A, Opt. 429	1	205,840	1	205,840	
Super Dome right chassis	A5202A, Opt. 429	1	218,435	1	218,435	
IPF Superdome Cell Board (sx1000)	A6866A	1	16,000	16	256,000	
3 Year Svc & Support Price (Hardware and Software)						838,197
4GB SDRAM (4x1GB DIMMS)	A6863A	1	17,750	64	1,136,000	
PCI-x I/O chassis	A6864A	1	16,805	12	201,660	
Core I/O Card	A6865A	1	1,045	1	1,045	
CPU Itanium 2, 1.5GHz w/6MB iL 3 cache (2 CPUs)	A6924A	1	40,000	32	1,280,000	
PCI 1000BT Lan Adapter	A6847A, Opt. 0D1	1	2,135	1	2,135	
I/O chassis enclosure for PCI chassis	A5862A	1	25,725	2	51,450	
Graphite I/O expansion power subsystem	A5861D	1	34,860	1	34,860	
PCI 2GB Fibre Channel Adapter	A6795A	1	2,240	72	161,280	
PCI Ultra160 SCSI Adapter	A6828A	1	1,323	1	1,323	
hp Surestore Disk System 2100	A5675A	1	700	1	700	
1-36GB LP 10K LVD SE U320 HDD	A6571A	1	720	4	2,880	
TA5300 Enclosure for DAT tape	C7508AZ	1	1,395	1	1,395	
DDS 4 tape	C5687B	1	1,450	1	1,450	
DVD Rom drive	C7499A	1	688	1	688	
SCSI Terminator LVD/SE HDTS68 Multimedia	C2364A	1	100	1	100	
HP Tape Array PSU/Fan Kit	C7496A	1	425	1	425	
SCSI Cable 10m VHDTS68/DHTS68 M/M Multimd	C2363B	1	335	1	335	
SCSI Cable 0.5m HDTS68 M/M Multimedia	C2978B	1	99	1	99	
SMS for HP Integrity Superdome Tower	A9801A	1	5,140	1	5,140	
				Subtotal	**3,563,240**	**838,197**
Server Software						
Oracle Database 10g Enerprise Edition for 3 years, Named User Plus		2	10,000	64	640,000	
Partitioning for 3 years, Named User Plus		2	2,500	64	160,000	
Oracle Database Server Support Package for 3 years:		2	6,000	1		6,000
HPUX 11i, V2 Foundation Operating Environment	B9429AC	1	2,370	64	151,680	
				Subtotal	**951,680**	**6,000**
Storage						
16 meter Fibre Optic Cable	A7525A	1	260	72	18,720	
Surestore VA 7110 w/ dual controller, 1024MB mem	A7294A	1	25,440	72	1,831,680	
*(large quantity discount)						
3 Year Support Price						273,168
73GB 15K RPM FC HDD (10% sparing included)*	A7288A, Opt 0D1	1	1,702	1188	2,022,392	103,680
*(large quantity discount)						
HP Rack System/E, 41U, Quartz Color	A4902A	1	1,910	6	11,460	
HP Rear Door for 41U Quartz Rack	A5213AZ	1	334	6	2,004	
Modular Power Dist.	A5137AZ	1	145	24	3,480	
200-240 Volts Power Option	A5137AZ, Opt AW4	1	94	24	2,256	
				Subtotal	**3,891,992**	**376,848**
				Total	**8,406,912**	**1,221,045**
Oracle Mandatory E-Business Discount on (Licenses and Support)					(161,200)	
Large Configuration Discount and Support Prepayment*					(3,952,665)	(592,022)
				Grand Total	**4,293,047**	**629,023**

Source: 1=HP,
2=Oracle (Pricing Contact: MaryBeth Pierantoni;
email: mary.beth.pierantori@oracle.com; phone number: 650-506-2118)

3-yr Cost of Ownership: **4,922,070**
QphH@3000GB: **45,247.8**
$/QphH@3000GB: $ **109**

*All discounts are based on US list prices and for similar quantities and configurations

Audited By: Francois Raab for InfoSizing (www.sizing.com)

Prices used in TPC benchmarks reflect 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 refelect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

HP Integrity Superdome Enterprise Server

TPC-H Rev 2.1.0

Report Date: December 31, 2003

Measurement Results

Database Scaling (SF/size)	3000
Total Data Storage/Database Size	26.33
Start of Database Load Time	2003-09-16 21:55:03
End of Database Load Time	2003-09-17 03:37:26
Database Load Time	05:42
Query Streams for Throughput Test (S)	8
TPC-H Power	54,559.8
TPC-H Throughput	37,525.2
TPC-H Composite Query-per-Hour Metric (QphH@3000GB)	45,247.8
Total System Price Over 3 Years	$4,922,070
TPC-H Price/Performance Metric ($/QphH@3000GB)	$109

Measurement Intervals

Measurement Interval in Throughput Test (Ts)	50,654

Duration of Stream Execution:

	SEED	Start Date/Time	End Date/Time	Duration
Stream 00	0917033726	09/17/2003 20:18:03	09/17/2003 22:25:57	2:07:54
Stream 01	0917033727	09/17/2003 22:26:45	09/18/2003 11:19:41	12:52:56
Stream 02	0917033728	09/17/2003 22:26:45	09/18/2003 11:31:0	13:04:15
Stream 03	0917033729	09/17/2003 22:26:45	09/18/2003 11:31:4	13:04:19
Stream 04	0917033730	09/17/2003 22:26:45	09/18/2003 10:34:17	12:07:32
Stream 05	0917033731	09/17/2003 22:26:45	09/18/2003 11:45:41	13:18:56
Stream 06	0917033732	09/17/2003 22:26:45	09/18/2003 11:43:41	13:16:56
Stream 07	0917033733	09/17/2003 22:26:45	09/18/2003 10:55:35	12:28:50
Stream 08	0917033734	09/17/2003 22:26:45	09/18/2003 10:54:41	12:27:56
Refresh		09/17/2003 22:26:45	09/18/2003 12:30:59	14:04:14

TPC-H Timing Intervals (in seconds)

Duration of stream execution:

Stream ID	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Stream 00	615.0	92.6	64.7	48.0	245.9	44.9	301.1	204.3	817.6	135.8	147.3	186.9
Stream 01	837.5	1250.5	295.3	73.8	1571.7	181.6	2867.7	1443.2	5654.7	570.6	573.8	918.2
Stream 02	2861.1	1736.0	323.0	268.0	1998.0	153.0	1333.2	953.5	5857.3	791.9	962.6	797.3
Stream 03	2304.1	1806.3	82.9	407.3	2435.8	111.7	2305.0	759.3	6031.2	1010.9	468.0	353.4
Stream 04	2521.4	1183.6	250.9	257.1	2158.0	64.7	2117.4	754.8	5236.2	571.4	498.7	702.2
Stream 05	2310.7	1482.1	400.8	248.1	1669.3	108.7	2366.0	1976.0	4635.0	923.3	615.6	1470.8
Stream 06	2235.6	1853.2	566.7	169.5	1855.8	324.5	2255.4	1565.7	6482.4	471.8	505.6	532.2
Stream 07	2036.3	1377.3	209.0	240.5	1920.2	185.7	2178.6	880.9	6123.5	635.3	678.2	1143.7
Stream 08	2535.92	2117.58	124.49	129.31	1657.61	56.24	2096.65	994.41	6227.88	602.68	665.28	1370.17
Minimum	837.5	1183.6	82.9	73.8	1571.7	56.2	1333.2	754.8	4635.0	471.8	468.0	353.4
Average	2205.3	1600.8	281.6	224.2	1908.3	148.3	2190.0	1166.0	5781.0	697.2	621.0	911.0
Maximum	2861.1	2117.6	566.7	407.3	2435.8	324.5	2867.7	1976.0	6482.4	1010.9	962.6	1470.8

Stream ID	Q13	Q14	Q15a	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream 00	549.5	40.0	49.5	147.3	318.0	1437.2	342.4	145.9	1054.5	117.3	241.0	327.8
Stream 01	2385.6	590.6	345.2	711.7	1370.8	12537.8	2540.7	724.3	8195.2	736.0	209.2	229.8
Stream 02	1675.9	324.5	338.7	1104.2	2206.7	13090.7	2445.9	455.8	6610.4	768.1	177.4	172.9
Stream 03	2310.3	253.2	237.6	236.6	2979.9	13034.8	1858.2	739.3	6369.6	964.2	148.0	169.1
Stream 04	1831.1	381.4	624.4	583.1	1376.4	13123.9	1505.5	1299.7	5866.9	743.8	142.0	171.3
Stream 05	2399.6	344.2	239.5	897.8	2045.6	13066.5	2194.4	1111.9	6568.6	861.9	137.8	173.9
Stream 06	2264.1	635.4	367.2	626.8	1799.1	12930.5	2637.5	1222.4	5601.9	913.0	176.4	173.3
Stream 08	2093.07	278.69	312.93	528.27	1972.36	10937.7	2860.68	292.18	6236.59	785.47	135.5	175.71
Stream 07	3030.0	325.0	342.4	792.1	1471.9	14848.2	1245.6	420.5	4222.6	622.5	147.3	178.1
Minimum	1675.9	253.2	237.6	236.6	1370.8	10937.7	1245.6	292.2	4222.6	622.5	135.5	169.1
Average	2248.7	391.6	351.0	685.1	1902.8	12946.3	2161.1	783.3	6209.0	799.4	159.2	180.5
Maximum	3030.0	635.4	624.4	1104.2	2979.9	14848.2	2860.7	1299.7	8195.2	964.2	209.2	229.8

Test Sponsors:

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

Juergen Mueller
Performance Manager BCS/ESTL
Hewlett-Packard
1911 Pruneridge Avenue, MS4105
Cupertino, CA 95014

September 24, 2003

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

Platform:	HP Integrity Superdome Enterprise Server
Database Manager:	Oracle Database 10g Enterprise Edition
Operating System:	HP-UX 11.i V2 64-bit

The results were:

CPU (Speed)	Memory	Disks	**QphH@3000GB**
HP Integrity Superdome Enterprise Server			
64 x Intanium2 (1.5 GHz)	6 MB L3-Cache/cpu 256 GB Main	1080 x 73 GB 4 x 36 GB	**45,247.8**

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

- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 3 TB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported

- The required ACID properties were verified and met
- The query input variables were generated by QGEN
- The query text was produced using minor modifications and the approved variant 15a
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 8 query streams
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified. A failure during the second run of the benchmark required the execution of a third run, from which the reported results were collected.
- 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

1 General Items

1.1 Benchmark Sponsor

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

Hewlett-Packard Company and Oracle are the test sponsors of this TPC Benchmark H benchmark.

1.2 Parameter Settings

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

Database Tuning Options

Optimizer/Query execution options

Query processing tool/language configuration parameters

Recovery/commit options

Consistency/locking options

Operating system and configuration parameters

Configuration parameters and options for any other software component incorporated into the pricing structure;

Compiler optimization options.

Appendix A contains the HP-UX and Oracle 10g Database Enterprise Edition with Partitioning parameters used in this benchmark.

1.3 Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

Measured Configuration:

- 64 1.5GHz Itanium2 CPUs each with 6MB L3 Cache.
- 256 GB Memory
- 72 PCI Fibre Channel 2X Cards
- 1 I/O Expansion Cabinet
- 1 HP 1000 BaseSX PCI Lan Adapter
- 72 SureStore VA7110 (with a total of 1080 73GB Disks)
- 1 High Availability Storage System (with a total of 4 9 GB 1-36GB LP 10K LVD SE U320 HDD Disks)
- 1 DVD ROM
- 1 SCSI Card

Priced Configuration:

- 64 1.5GHz Itanium2 CPUs each with 6MB L3 Cache.
- 256 GB Memory
- 72 PCI Fibre Channel 2X Card
- 1 I/O Expansion Cabinet
- 1 HP 1000 BaseSX PCI Lan Adapter

- 72 SureStore VA7110 (with a total of 1080 73GB Disks)
- 1 hp surestore disk system 2100 (with a total of 4 36 GB 1-36GB LP 10K LVD SE U320 HDD Disks)
- 1 DVD ROM
- 1 SCSI Card

The difference between measured and priced is a High Availability Storage System for the root disk which currently is obsolete. For the priced system we used a Surestore Disk System 2100 was substituted.

Terminal

Keyboard

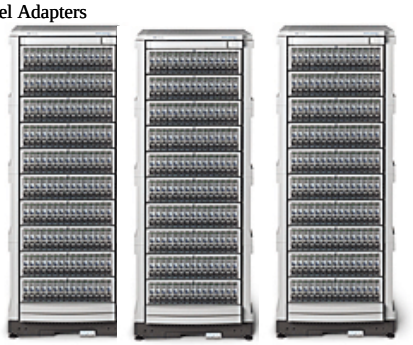
Mouse

Measured Configuration

72 HP Surestore Virtual Array 7110

- 1080 73GB 15K RPM Disk Drives

72 Fiber Channel Adapters



HP Integrity Superdome Enterprise Server

84 Fibre Channel Adapter Cards

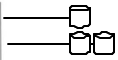
64 – 1.5GHz Itanium2 Processors

256GB Memory

6MB iL3 Cache

Additional I/O Chassis

SCSI



High Availability Storage System

3 9.1GB LVD Ultra2-SCSI Disks

LAN

SCSI

HP DVD-ROM

Priced Configuration

Terminal

Keyboard

Mouse

HP Integrity Superdome Enterprise Server

64 – 1.5GHz Itanium2 Processors

256GB Memory

6MB iL3 Cache

LAN

SCSI

SCSI

HP DVD-ROM

72 HP Surestore Virtual Array 7100

- 1080 73GB 15K RPM Disk Drives

72 Fiber Channel Adapters

Additional I/O Chassis

1 hp surestore disk system 2100

With 3 -36GB LP 10K
LVD SE U320 HDD

2 Clause 1 Logical Database Design Related Items

2.1 Database Definition Statements

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

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

2.2 Physical Organization

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

No record clustering or index clustering was used. Columns were reordered in the tables – please refer to the table create statements for the ordering.

2.3 Horizontal Partitioning

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

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

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

2.4 Replication

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

No replication was used.

3 Clause 2 Queries and Refresh Functions

3.1 Query Language

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

SQL was the query language used to implement all queries.

3.2 Verifying Method for Random Number Generation

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

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

3.3 Generating Values for Substitution Parameters

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

QGEN version 1.3.0 was used to generate the substitution parameters.

3.4 Query Text and Output Data from Qualification Database

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

- Appendix C contains the actual query text and query output.

3.5 Query Substitution Parameters and Seeds Used

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

Appendix E contains the seed and query substitution parameters.

3.6 Query Isolation Level

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

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

3.7 Source Code of Refresh Functions

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

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

4 Clause 3 Database System Properties

4.1 ACID Properties

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

Source code for ACID test is included in Appendix C.

4.2 Atomicity

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

Completed Transaction

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

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

Aborted Transaction

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

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

4.3 Consistency

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

Consistency Test

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

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

4.4 Isolation

Operations of concurrent transactions must yield results, which are indistinguishable from the results, which would be obtained by forcing each transaction to be serially executed to completion in 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. An ACID Query was started for the same O_KEY used in step 1. The ACID Query blocked and did not see any uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was resumed, and COMMITTED.
4. The ACID Query completed. It returned the data as committed by the ACID Transaction.

Read-Write Conflict with Rollback

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

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

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

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

Concurrent Progress of Read and Write on Different Tables

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

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

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

Read-Only Query Conflict with Update Transactions

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

4.5 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 3.5.3.

Failure of a Durable Medium

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

The disks containing TPC-H tables and log files were on RAID 1/0 protected disk groups. During the durability test, one disk was removed from each RAID group containing the data and the log. The test continued uninterrupted, because of the RAID protection.

System Crash

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

The system crash and memory failure tests were combined. Power to the servers was turned off during the 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.

5 Clause 4 Scaling and Database Population

5.1 Ending Cardinality of Tables

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

Table	Cardinality
ORDER	4,500,000,000
LINEITEM	18,000,048,306
CUSTOMER	450,000,000
PART	600,000,000
SUPPLIER	30,000,000
PARTSUPP	2,400,000,000
NATION	25
REGION	5

5.2 Distribution of Tables and Logs Across Media

On each VA7100 array, several LUNs were created in RAID 1/0 mode.

These LUNs were for:

- flat-files
- swap
- lineitem + orders table
- part, supp, partsupp, nation, region tables and all indexes
- temp
- misc (logs/sys/default, etc).

24 logical volumes were created across the 72 arrays with 3 LUNs in each volume group. Each of these volume groups was divided into 28 lvols. For lineitem table, 14 of these lvols from each of the 24 volume groups was used as an independent Oracle tablespace (336 in all). Similarly, for orders table, 14 of these lvols from each of the 24 volume groups was used as an independent oracle tablespace (336 in all).

16 logical volumes were created across 64 arrays with 4 LUNs in each volume group. The other tables and indexes were created evenly across these logical volumes.

8 logical volumes were created across 72 arrays with 9 LUNs in each volume group. The temp files were created evenly across these logical volumes.

A single volume group was created across all 72 arrays and sys/default/undo/logs were created in this volume group.

Another single volume group was created across all 72 arrays for use as swap.

OS root disk was configured on one disk from the JBOD array.

5.3 Database Partition/Replication Mapping

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

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

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

5.4 RAID Feature

Implementation 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.

RAID1/0 was used for log, data, temp, index, and all storage disks.

5.5 DBGEN Modification

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

The supplied DBGEN version 1.3.0 was not modified to generate the database population for this benchmark.

5.6 Database Load Time

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

The database load time was 05:42

5.7 Data Storage Ratio

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

The data storage ratio is computed from the following information:

Type	Quantity	Disk Size	Total
1 hp surestore disk system 2100	4	36	144
72 SureStore VA7110	1080	73	78,840.0
TOTAL			**78,984.0**
Scale Factor			**3,000**
Storage Ratio			**26.33**

5.8 Database Load Mechanism Details and Illustration

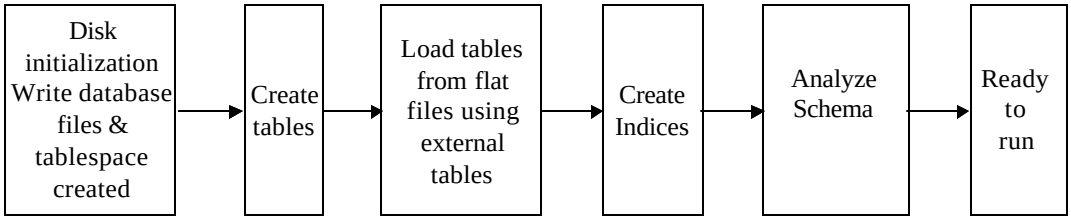
The details of the database load must be described, including a block diagram illustrating the overall process.

The database was loaded using data generation stored on the flat files all on the tested and priced configuration

5.9 Qualification Database Configuration

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

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



6 Clause 5 Performance Metrics and Execution-Rules

6.1 System Activity Between Load and Performance Tests

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

A script was run to display the hardware configurations of the SUT.

Auditor requested queries were run against the database to verify the correctness of the database load.

The database was restarted.

All scripts and queries used are included in Appendix E.

6.2 Steps in the Power Test

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

The following steps were used to implement the power test:

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

6.3 Timing Intervals for Each Query and Refresh Functions

The timing intervals for each query for both refresh functions must be reported for the power test.

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

6.4 Number of Streams for the Throughput Test

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

8 streams were used for the throughput test.

6.5 Start and End Date/Time of Each Query Stream

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

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

6.6 Total Elapsed Time of the Measurement Interval

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

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

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

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

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

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

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

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

6.9 Performance Metrics

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

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

6.10 The Performance Metric and Numerical Quantities from Both Runs

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

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

	QppH@3000GB	QthH@3000GB	QphH@3000GB
Reported Run	54,559.8	37,525.2	45,247.8
Reproducibility Run	58,652.0	37,216.6	46,720.7
% Difference	7.5%	0.8%	3.3%

6.11 System Activity Between Performance Tests

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

The database was restarted between the two runs.

7 Clause 6 SUT and Driver Implementation Related Items

7.1 Driver

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

All stream executions are performed by a single script. QGEN is used to produce query text.

For each power-test run:

- The SQL for RF1 is submitted to the database
- Then the queries as generated by QGEN are submitted in the order defined by Clause 5.3.5.4
- The SQL for RF2 is submitted to the database**.**

7.2 Implementation-Specific Layer (ISL)

If an implementation specific layer is used, then a detailed description of how it performs its functions must be provided. All related source code, scripts and configuration files must be disclosed. The information provided should be sufficient for an independent reconstruction of the implementation specific layer.

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

7.3 Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2. is used, such use must be disclosed..

Profile-directed optimization subject to the requirements of *5.2.9* and *5.2.10* was not used.

8 Clause 7 Pricing

8.1 Hardware and Software Used in the Priced System

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

A detailed list of hardware and software used in the priced system is included in the pricing sheet in the executive summary. All prices are currently effective.

8.2 Total Three Year Price

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

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

8.3 Availability Date

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

Availability Dates:

Server Hardware	Now
Server Software	Now
Storage	Available Now
Database Manager (Oracle 10g Database Enterprise Edition with Partitioning)	March 25, 2004

9 Clause 8 Auditor's Information and Attestation Letter

9.1 Auditor's Report

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

This implementation of the TPC Benchmark H was audited by Francois Raab for InfoSizing Further information regarding the audit process may be obtained from:

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

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

10 Report Availability

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

Transaction Processing Performance Council
404 Balboa Street
San Francisco, CA 94118
Voice: 415-750-8260
Fax: 415-751-4829

or your local Hewlett-Packard sales office

Appendix A Parameter Settings

A.1 3TB-run.ora

```
hpux_sched_noage              = 180
statistics_level              = BASIC
aq_tm_processes               = 0
audit_trail                   = FALSE
compatible                    = 10.0.0.0
control_files                 =
(/oracle/dbs/control1,/oracle/dbs/control2)
db_block_checksum             = false
db_block_size                 = 8192
db_cache_size                 = 12g
db_file_multiblock_read_count = 256
db_files                      = 1200
db_name                       = 3tb
db_writer_processes           = 8
dml_locks                     = 40000
enqueue_resources             = 40000
global_names                  = FALSE
instance_name                 = tpch
large_pool_size               = 3g
log_buffer                    = 33554432
log_checkpoints_to_alert      = true
max_dump_file_size            = unlimited
nls_date_format               = YYYY-MM-DD
open_cursors                  = 1024
optimizer_features_enable     = 10.0.0.1
optimizer_mode                = CHOOSE
parallel_adaptive_multi_user  = TRUE
parallel_execution_message_size = 16384
parallel_max_servers          = 1024
parallel_min_servers          = 1024
pga_aggregate_target          = 70g
processes                     = 5000
query_rewrite_enabled         = true
recovery_parallelism          = 32
replication_dependency_tracking = false
shared_pool_size              = 1073741824
transactions                  = 10
undo_management               = auto
undo_retention                = 200000
```

A.2 system

```
*
* Created on Sat Sep 13 22:27:47 2003
*
version 1
configuration nextboot "imported from system_tpch" [3f63fc4f]
*
* Module entries
*
module ipf loaded 0.1.0
module mpt best [3F1DCA3B]
module vols best [3F1DAFDA]
module vol best [3F1DAFDA]
module vxdmp best [3F1DAFD9]
module vxvm best [3F1DAFDA]
module lv best [3F237870]
module lvm best [3F237870]
module vxportal best [3F237870]
module vxfs best [3F237870]
module pfil auto 0.1.0
module igelan best [3F17DABC]
module iether best [3F17DC0F]
module gelan best [3F17D80A]
module fddi4 best [3F16C8EB]
module td best [3F1D1375]
module cifs best [3F1C3A86]
module pckt best [3F237870]
module ptm best [3F237870]
module pts best [3F237870]
module ptem best [3F237870]
module ldterm best [3F237870]
module ffs best [3F237870]
module pipemod best [3F237870]
module pipedev best [3F237870]
module tirdwr best [3F237870]
module timod best [3F237870]
module sc best [3F237870]
module echo best [3F237870]
module sad best [3F237870]
module strlog best [3F237870]
module clone best [3F237870]
module hpstreams best [3F237870]
module cachefsc best [3F237870]
module autofsc best [3F237870]
module rpcmod best [3F237870]
module nfsm best [3F237870]
module nfs_client best [3F237870]
module nfs_server best [3F237870]
module nfs_core best [3F237870]
module nms best [3F237870]
module netdiag1 best [3F237870]
module token_arp best [3F237870]
module dlpi best [3F237870]
module intl100 best [3F237870]
module btlan best [3F237870]
module tels best [3F237870]
module telm best [3F237870]
module tun best [3F237870]
module uipc best [3F237870]
module inet best [3F237870]
module rng loaded 0.1.0
module cdfs best 0.1.0
module dev_config best [3F237870]
module dmem best [3F237870]
module diag2 best [3F237870]
module c8xx best [3F237870]
module pdh best [3F237870]
module lion_psm best [3F237870]
module ia64_psm best [3F237870]
module wxb_hp best [3F237870]
module sac best [3F237870]
module acpi_node best [3F237870]
module LCentIf best [3F237870]
module ipmi best [3F237870]
module pty1 best [3F237870]
module pty0 best [3F237870]
module azusa_psm best [3F237870]
module sctl best [3F237870]
module sdisk best [3F237870]
module tgt best [3F237870]
module asio0 best [3F237870]
module lba best [3F237870]
module sba best [3F237870]
module cell best [3F237870]
module root best [3F237870]
*
* Swap entries
*
```

```
*
* Dump entries
*
dump lvol
*
* Driver binding entries
*
*
* Tunables entries
*
tunable unlockable_mem 1
tunable timezone 480
tunable maxswapchunks 65536
tunable swchunk 65536
tunable nhtbl_scale 1
tunable max_async_ports 2048
tunable maxusers 32
tunable eqmemsize 512
tunable bufpages 50000
tunable maxtsiz_64bit 4294967296
tunable maxssiz_64bit 1073741824
tunable maxdsiz_64bit 0x80000000
tunable maxdsiz 0x40000000
tunable shmmax 0x40000000000
tunable semvmx 32768
tunable nproc 4096
tunable nfile 500000
tunable msgmni 512
tunable maxfiles_lim 4096
tunable maxfiles 4096
tunable swapmem_on 0
tunable vx_fancyra_enable 1
tunable vxfs_max_ra_kbytes 1024
tunable vxfs_ra_per_disk 1024
tunable dbc_max_pct 3
tunable dbc_min_pct 3
tunable vps_ceiling 64
tunable STRMSGSZ 65535
tunable shmseg 512
tunable shmmni 2048
tunable semume 512
tunable semmnu 4092
tunable semmns 8192
tunable semmni 4096
tunable nswapdev 25
tunable npty 200
tunable ninode 120000
tunable msgtql 5120
tunable msgssz 128
tunable msgseg 20480
tunable msgmnb 65536
tunable msgmax 32768
tunable msgmap 5122
tunable maxvgs 99
tunable maxuprc 3277
tunable maxtsiz 1073741824
tunable maxssiz 0x10000000
tunable max_thread_proc 2048
tunable hfs_revra_per_disk 256
tunable hfs_ra_per_disk 256
tunable hfs_max_revra_blocks 20
tunable hfs_max_ra_blocks 20
tunable create_fastlinks 1
tunable nstrpty 200
tunable vxfs_ifree_timelag 3600000
```

A.3 env

```
########## MACHINE PARAMETERS
####################
#export RAC_NODES="mach1 mach2 mach3 mach4"
########## PATHS ##################################
export KIT_DIR=/dbms/oracle10i/kit
export SCHEMA_DIR=$KIT_DIR/schema
export PERL=/opt/perl/bin/perl
export BUMPX_DIR=$KIT_DIR/bumpx
export BUMPX_OUT=$KIT_DIR/bumpx
export UTILS=$KIT_DIR/utils
export TEST_DB=/tmp
export QUAL_DB=$TEST_DB
export DBGEN=$KIT_DIR/dbgen
export ACID_DIR=$KIT_DIR/acid
export QEXEC=$KIT_DIR/utils
export QUERIES=$KIT_DIR/queries
export ANSWERS=$KIT_DIR/answers
export ANS2VAL=/dbms/oracle10i/kit/acid/answers2validate
export ACID_OUT=$KIT_DIR/out
export DSS_CONFIG=$DBGEN
export DSS_QUERY=$KIT_DIR/queries
export DSS_PATH=$ADE_VIEW_ROOT
export MAINT=$KIT_DIR/maintenance
export CC=/opt/ansic/bin/cc
export FRAME=$KIT_DIR/frame
export REGR_TEST=$KIT_DIR/internal/regression_test
export SCALE_FACTOR=3000
export UPDATE_1_DOP=32
export UPDATE_2_DOP=128
############ FRAME STUFF
export FRAME_PATH=$KIT_DIR/frame

#export ORACORE3INCL=/vobs/oracore3/include
#export ORACORE3PUBL=/vobs/oracore3/public
export ORACORE3INCL=$ORACLE_HOME/rdbms/demo
export ORACORE3PUBL=$ORACLE_HOME/rdbms/public
#export RDBMSPUBL=/vobs/rdbms/public
export RDBMSPUBL=$ORACLE_HOME/rdbms/public
#export NETWORKPUBL=/vobs/network_src/public
export NETWORKPUBL=$ORACLE_HOME/network/public
export RDBMSDEMO=$ORACLE_HOME/rdbms/demo
export PLSQLDEMO=$ORACLE_HOME/plsql/demo
export PLSQLPUBL=$ORACLE_HOME/plsql/public
export O=$ORACLE_HOME
export
PATH=./:${BUMPX_DIR}:${UTILS}:${DBGEN}:${MAINT}:${
ACID_DIR}:${FRAME}/bin:${FRAME}/bin:${REGR_TEST}:${
PATH}
#
########## ENVIRONMENT VARIABLES
#################
export WORKLOAD=TPCH
export HOST=
#export OPTLEVEL=X02
export GETOPT=-DSTDLIB_HAS_GETOPT
export PLATFORM=
#export INITORA=$KIT_DIR/schema/test_db/testdb.ora
#export INITORA=$KIT_DIR/schema/test_db/sf100.ora

########## ALIASES ################################

########## RULES - do not change these ############
case "$SCALE_FACTOR" in
  1) export NUM_STREAMS=2;;
  10) export NUM_STREAMS=3;;
  100) export NUM_STREAMS=4;;
  300) export NUM_STREAMS=6;;
```

```
  1000) export NUM_STREAMS=7;;
  3000) export NUM_STREAMS=8;;
  10000) export NUM_STREAMS=9;;
esac
DATABASE_USER=tpch/tpch
```

Appendix B Build Programs and Scripts

B.1 3TB.DAT

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

%b-preproc

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


*load1
\sqlldr {}

*mknod
\mknod {}

*dbgen
\dbgen {}

*sh
\{}

%e-preproc
%b-dbcre
*bgon=1
#########################################################
#####################
# Database Creation Phase
*sql
{
shutdown abort;
}
*wait
# creating database  and undo tablespace
*sql
{
startup pfile=/oracle/dbs/3TB_run.ora nomount;
create database
  controlfile reuse
  logfile '/dbms/links/log_1' size 22000m reuse,
          '/dbms/links/log_2' size 20000m reuse
  datafile '/dbms/links/sys_1' size 2000m reuse
  sysaux datafile '/dbms/links/aux' size 2000m reuse
  undo tablespace ts_undo
       datafile  '/dbms/links/undo_1' size 22000m reuse
  maxdatafiles 1000
  maxinstances 1
;
}
*wait
# building data dictionary
*sql
{
set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@?/sqlplus/admin/pupbld.sql;
spool off
}
*wait
*bgoff
%e-dbcre
%b-sctso
*bgon=64
#########################################################
#####################
# Schema Creation Phase - datafiles only (no tables or users)
# creating data tablespaces, datafiles
# creating tpch's ts_one tablespace

*sql
{
--drop tablespace ts_default including contents;
create tablespace ts_default
datafile '/dbms/links/default_1' size 4000m reuse
extent management local
autoallocate
;
}
*sql
{
--drop tablespace ts_l1 including contents;
create tablespace ts_l1
datafile '/dbms/links/line_1' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l2 including contents;
create tablespace ts_l2
datafile '/dbms/links/line_2' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l3 including contents;
create tablespace ts_l3
datafile '/dbms/links/line_3' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l4 including contents;
create tablespace ts_l4
datafile '/dbms/links/line_4' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l5 including contents;
create tablespace ts_l5
datafile '/dbms/links/line_5' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l6 including contents;
create tablespace ts_l6
datafile '/dbms/links/line_6' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l7 including contents;
create tablespace ts_l7
datafile '/dbms/links/line_7' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l8 including contents;
create tablespace ts_l8
datafile '/dbms/links/line_8' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l9 including contents;
create tablespace ts_l9
datafile '/dbms/links/line_9' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l10 including contents;
create tablespace ts_l10
datafile '/dbms/links/line_10' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l11 including contents;
create tablespace ts_l11
datafile '/dbms/links/line_11' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l12 including contents;
create tablespace ts_l12
datafile '/dbms/links/line_12' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l13 including contents;
create tablespace ts_l13
datafile '/dbms/links/line_13' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l14 including contents;
create tablespace ts_l14
datafile '/dbms/links/line_14' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l15 including contents;
create tablespace ts_l15
datafile '/dbms/links/line_15' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l16 including contents;
create tablespace ts_l16
datafile '/dbms/links/line_16' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l17 including contents;
create tablespace ts_l17
datafile '/dbms/links/line_17' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l18 including contents;
create tablespace ts_l18
datafile '/dbms/links/line_18' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l19 including contents;
create tablespace ts_l19
datafile '/dbms/links/line_19' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l20 including contents;
```

```
create tablespace ts_l20
datafile '/dbms/links/line_20' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l21 including contents;
create tablespace ts_l21
datafile '/dbms/links/line_21' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l22 including contents;
create tablespace ts_l22
datafile '/dbms/links/line_22' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l23 including contents;
create tablespace ts_l23
datafile '/dbms/links/line_23' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l24 including contents;
create tablespace ts_l24
datafile '/dbms/links/line_24' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l25 including contents;
create tablespace ts_l25
datafile '/dbms/links/line_25' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l26 including contents;
create tablespace ts_l26
datafile '/dbms/links/line_26' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l27 including contents;
create tablespace ts_l27
datafile '/dbms/links/line_27' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l28 including contents;
create tablespace ts_l28
datafile '/dbms/links/line_28' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l29 including contents;
create tablespace ts_l29
datafile '/dbms/links/line_29' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlim ited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l30 including contents;
create tablespace ts_l30
datafile '/dbms/links/line_30' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l31 including contents;
create tablespace ts_l31
datafile '/dbms/links/line_31' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l32 including contents;
create tablespace ts_l32
datafile '/dbms/links/line_32' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l33 including contents;
create tablespace ts_l33
datafile '/dbms/links/line_33' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l34 including contents;
create tablespace ts_l34
datafile '/dbms/links/line_34' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l35 including contents;
create tablespace ts_l35
datafile '/dbms/links/line_35' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l36 including contents;
create tablespace ts_l36
datafile '/dbms/links/line_36' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l37 including contents;
create tablespace ts_l37
datafile '/dbms/links/line_37' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l38 including contents;
create tablespace ts_l38
datafile '/dbms/links/line_38' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l39 including contents;
create tablespace ts_l39
datafile '/dbms/links/line_39' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l40 including contents;
create tablespace ts_l40
datafile '/dbms/links/line_40' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l41 including contents;
create tablespace ts_l41
datafile '/dbms/links/line_41' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l42 including contents;
create tablespace ts_l42
datafile '/dbms/links/line_42' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l43 including contents;
create tablespace ts_l43
datafile '/dbms/links/line_43' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l44 including contents;
create tablespace ts_l44
datafile '/dbms/links/line_44' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l45 including contents;
create tablespace ts_l45
datafile '/dbms/links/line_45' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l46 including contents;
create tablespace ts_l46
datafile '/dbms/links/line_46' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l47 including contents;
create tablespace ts_l47
datafile '/dbms/links/line_47' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l48 including contents;
create tablespace ts_l48
datafile '/dbms/links/line_48' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l49 including contents;
create tablespace ts_l49
datafile '/dbms/links/line_49' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l50 including contents;
create tablespace ts_l50
datafile '/dbms/links/line_50' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l51 including contents;
```

```
create tablespace ts_l51
datafile '/dbms/links/line_51' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l52 including contents;
create tablespace ts_l52
datafile '/dbms/links/line_52' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l53 including contents;
create tablespace ts_l53
datafile '/dbms/links/line_53' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l54 including contents;
create tablespace ts_l54
datafile '/dbms/links/line_54' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l55 including contents;
create tablespace ts_l55
datafile '/dbms/links/line_55' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l56 including contents;
create tablespace ts_l56
datafile '/dbms/links/line_56' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l57 including contents;
create tablespace ts_l57
datafile '/dbms/links/line_57' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l58 including contents;
create tablespace ts_l58
datafile '/dbms/links/line_58' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l59 including contents;
create tablespace ts_l59
datafile '/dbms/links/line_59' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l60 including contents;
create tablespace ts_l60
datafile '/dbms/links/line_60' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l61 including contents;
create tablespace ts_l61
datafile '/dbms/links/line_61' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l62 including contents;
create tablespace ts_l62
datafile '/dbms/links/line_62' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l63 including contents;
create tablespace ts_l63
datafile '/dbms/links/line_63' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l64 including contents;
create tablespace ts_l64
datafile '/dbms/links/line_64' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l65 including contents;
create tablespace ts_l65
datafile '/dbms/links/line_65' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l66 including contents;
create tablespace ts_l66
datafile '/dbms/links/line_66' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l67 including contents;
create tablespace ts_l67
datafile '/dbms/links/line_67' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l68 including contents;
create tablespace ts_l68
datafile '/dbms/links/line_68' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l69 including contents;
create tablespace ts_l69
datafile '/dbms/links/line_69' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l70 including contents;
create tablespace ts_l70
datafile '/dbms/links/line_70' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l71 including contents;
create tablespace ts_l71
datafile '/dbms/links/line_71' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l72 including contents;
create tablespace ts_l72
datafile '/dbms/links/line_72' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l73 including contents;
create tablespace ts_l73
datafile '/dbms/links/line_73' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l74 including contents;
create tablespace ts_l74
datafile '/dbms/links/line_74' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l75 including contents;
create tablespace ts_l75
datafile '/dbms/links/line_75' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l76 including contents;
create tablespace ts_l76
datafile '/dbms/links/line_76' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l77 including contents;
create tablespace ts_l77
datafile '/dbms/links/line_77' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l78 including contents;
create tablespace ts_l78
datafile '/dbms/links/line_78' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l79 including contents;
create tablespace ts_l79
datafile '/dbms/links/line_79' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l80 including contents;
create tablespace ts_l80
datafile '/dbms/links/line_80' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l81 including contents;
create tablespace ts_l81
datafile '/dbms/links/line_81' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l82 including contents;
```

```
create tablespace ts_l82
datafile '/dbms/links/line_82' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l83 including contents;
create tablespace ts_l83
datafile '/dbms/links/line_83' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l84 including contents;
create tablespace ts_l84
datafile '/dbms/links/line_84' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l85 including contents;
create tablespace ts_l85
datafile '/dbms/links/line_85' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l86 including contents;
create tablespace ts_l86
datafile '/dbms/links/line_86' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l87 including contents;
create tablespace ts_l87
datafile '/dbms/links/line_87' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l88 including contents;
create tablespace ts_l88
datafile '/dbms/links/line_88' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l89 including contents;
create tablespace ts_l89
datafile '/dbms/links/line_89' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l90 including contents;
create tablespace ts_l90
datafile '/dbms/links/line_90' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l91 including contents;
create tablespace ts_l91
datafile '/dbms/links/line_91' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l92 including contents;
create tablespace ts_l92
datafile '/dbms/links/line_92' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l93 including contents;
create tablespace ts_l93
datafile '/dbms/links/line_93' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l94 including contents;
create tablespace ts_l94
datafile '/dbms/links/line_94' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l95 including contents;
create tablespace ts_l95
datafile '/dbms/links/line_95' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l96 including contents;
create tablespace ts_l96
datafile '/dbms/links/line_96' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l97 including contents;
create tablespace ts_l97
datafile '/dbms/links/line_97' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l98 including contents;
create tablespace ts_l98
datafile '/dbms/links/line_98' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l99 including contents;
create tablespace ts_l99
datafile '/dbms/links/line_99' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l100 including contents;
create tablespace ts_l100
datafile '/dbms/links/line_100' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l101 including contents;
create tablespace ts_l101
datafile '/dbms/links/line_101' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l102 including contents;
create tablespace ts_l102
datafile '/dbms/links/line_102' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l103 including contents;
create tablespace ts_l103
datafile '/dbms/links/line_103' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l104 including contents;
create tablespace ts_l104
datafile '/dbms/links/line_104' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l105 including contents;
create tablespace ts_l105
datafile '/dbms/links/line_105' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l106 including contents;
create tablespace ts_l106
datafile '/dbms/links/line_106' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l107 including contents;
create tablespace ts_l107
datafile '/dbms/links/line_107' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l108 including contents;
create tablespace ts_l108
datafile '/dbms/links/line_108' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l109 including contents;
create tablespace ts_l109
datafile '/dbms/links/line_109' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l110 including contents;
create tablespace ts_l110
datafile '/dbms/links/line_110' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l111 including contents;
create tablespace ts_l111
datafile '/dbms/links/line_111' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l112 including contents;
create tablespace ts_l112
datafile '/dbms/links/line_112' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l113 including contents;
```

```
create tablespace ts_l113
datafile '/dbms/links/line_113' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l114 including contents;
create tablespace ts_l114
datafile '/dbms/links/line_114' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l115 including contents;
create tablespace ts_l115
datafile '/dbms/links/line_115' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l116 including contents;
create tablespace ts_l116
datafile '/dbms/links/line_116' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l117 including contents;
create tablespace ts_l117
datafile '/dbms/links/line_117' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l118 including contents;
create tablespace ts_l118
datafile '/dbms/links/line_118' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l119 including contents;
create tablespace ts_l119
datafile '/dbms/links/line_119' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l120 including contents;
create tablespace ts_l120
datafile '/dbms/links/line_120' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l121 including contents;
create tablespace ts_l121
datafile '/dbms/links/line_121' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l122 including contents;
create tablespace ts_l122
datafile '/dbms/links/line_122' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l123 including contents;
create tablespace ts_l123
datafile '/dbms/links/line_123' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l124 including contents;
create tablespace ts_l124
datafile '/dbms/links/line_124' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l125 including contents;
create tablespace ts_l125
datafile '/dbms/links/line_125' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l126 including contents;
create tablespace ts_l126
datafile '/dbms/links/line_126' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l127 including contents;
create tablespace ts_l127
datafile '/dbms/links/line_127' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l128 including contents;
create tablespace ts_l128
datafile '/dbms/links/line_128' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l129 including contents;
create tablespace ts_l129
datafile '/dbms/links/line_129' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l130 including contents;
create tablespace ts_l130
datafile '/dbms/links/line_130' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l131 including contents;
create tablespace ts_l131
datafile '/dbms/links/line_131' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l132 including contents;
create tablespace ts_l132
datafile '/dbms/links/line_132' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l133 including contents;
create tablespace ts_l133
datafile '/dbms/links/line_133' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l134 including contents;
create tablespace ts_l134
datafile '/dbms/links/line_134' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l135 including contents;
create tablespace ts_l135
datafile '/dbms/links/line_135' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l136 including contents;
create tablespace ts_l136
datafile '/dbms/links/line_136' size 9000m reuse
extent management dictio nary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l137 including contents;
create tablespace ts_l137
datafile '/dbms/links/line_137' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l138 including contents;
create tablespace ts_l138
datafile '/dbms/links/line_138' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l139 including contents;
create tablespace ts_l139
datafile '/dbms/links/line_139' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l140 including contents;
create tablespace ts_l140
datafile '/dbms/links/line_140' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l141 including contents;
create tablespace ts_l141
datafile '/dbms/links/line_141' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l142 including contents;
create tablespace ts_l142
datafile '/dbms/links/line_142' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l143 including contents;
create tablespace ts_l143
datafile '/dbms/links/line_143' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l144 including contents;
```

```
create tablespace ts_l144
datafile '/dbms/links/line_144' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l145 including contents;
create tablesp ace ts_l145
datafile '/dbms/links/line_145' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l146 including contents;
create tablespace ts_l146
datafile '/dbms/links/line_146' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l147 including contents;
create tablespace ts_l147
datafile '/dbms/links/line_147' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l148 including contents;
create tablespace ts_l148
datafile '/dbms/links/line_148' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l149 including contents;
create tablespace ts_l149
datafile '/dbms/links/line_149' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l150 including contents;
create tablespace ts_l150
datafile '/dbms/links/line_150' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l151 including contents;
create tablespace ts_l151
datafile '/dbms/links/line_151' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l152 including contents;
create tablespace ts_l152
datafile '/dbms/links/line_152' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m max extents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l153 including contents;
create tablespace ts_l153
datafile '/dbms/links/line_153' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l154 including contents;
create tablespace ts_l154
datafile '/dbms/links/line_154' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l155 including contents;
create tablespace ts_l155
datafile '/dbms/links/line_155' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l156 including contents;
create tablespace ts_l156
datafile '/dbms/links/line_156' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l157 including contents;
create tablespace ts_l157
datafile '/dbms/links/line_157' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l158 including contents;
create tablespace ts_l158
datafile '/dbms/links/line_158' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l159 including contents;
create tablespace ts_l159
datafile '/dbms/links/line_159' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l160 including contents;
create tablespace ts_l160
datafile '/dbms/links/line_160' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l161 including contents;
create tablespace ts_l161
datafile '/dbms/links/line_161' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l162 including contents;
create tablespace ts_l162
datafile '/dbms/links/line_162' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l163 including contents;
create tablespace ts_l163
datafile '/dbms/links/line_163' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l164 including contents;
create tablespace ts_l164
datafile '/dbms/links/line_164' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l165 including contents;
create tablespace ts_l165
datafile '/dbms/links/line_165' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l166 including contents;
create tablespace ts_l166
datafile '/dbms/links/line_166' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l167 including contents;
create tablespace ts_l167
datafile '/dbms/links/line_167' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l168 including contents;
create tablespace ts_l168
datafile '/dbms/links/line_168' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l169 including contents;
create tablespace ts_l169
datafile '/dbms/links/line_169' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l1 70 including contents;
create tablespace ts_l170
datafile '/dbms/links/line_170' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l171 including contents;
create tablespace ts_l171
datafile '/dbms/links/line_171' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l172 including contents;
create tablespace ts_l172
datafile '/dbms/links/line_172' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l173 including contents;
create tablespace ts_l173
datafile '/dbms/links/line_173' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l174 including contents;
create tablespace ts_l174
datafile '/dbms/links/line_174' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l175 including contents;
```

```
create tablespace ts_l175
datafile '/dbms/links/line_175' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l176 including contents;
create tablespace ts_l176
datafile '/dbms/links/line_176' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l177 including contents;
create tablespace ts_l177
datafile '/dbms/links/line_177' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l178 including contents;
create tablespace ts_l178
datafile '/dbms/links/line_178' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l179 including contents;
create tablespace ts_l179
datafile '/dbms/links/line_179' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l180 including contents;
create tablespace ts_l180
datafile '/dbms/links/line_180' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l181 including contents;
create tablespace ts_l181
datafile '/dbms/links/line_181' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l182 including contents;
create tablespace ts_l182
datafile '/dbms/links/line_182' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l183 including contents;
create tablespace ts_l183
datafile '/dbms/links/line_183' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l184 including contents;
create tablespace ts_l184
datafile '/dbms/links/line_184' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l185 including contents;
create tablespace ts_l185
datafile '/dbms/links/line_185' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l186 including contents;
create tablespace ts_l186
datafile '/dbms/links/line_186' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l187 including contents;
create tablespace ts_l187
datafile '/dbms/links/line_187' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l188 including contents;
create tablespace ts_l188
datafile '/dbms/links/line_188' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l189 including contents;
create tablespace ts_l189
datafile '/dbms/links/line_189' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l190 including contents;
create tablespace ts_l190
datafile '/dbms/links/line_190' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l191 including contents;
create tablespace ts_l191
datafile '/dbms/links/line_191' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l192 including contents;
create tablespace ts_l192
datafile '/dbms/links/line_192' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l193 including contents;
create tablespace ts_l193
datafile '/dbms/links/line_193' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l194 including contents;
create tablespace ts_l194
datafile '/dbms/links/line_194' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l195 including contents;
create tablespace ts_l195
datafile '/dbms/links/line_195' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l196 including contents;
create tablespace ts_l196
datafile '/dbms/links/line_196' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l197 including contents;
create tablespace ts_l197
datafile '/dbms/links/line_197' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l198 including contents;
create tablespace ts_l198
datafile '/dbms/links/line_198' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l199 including contents;
create tablespace ts_l199
datafile '/dbms/links/line_199' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l200 including contents;
create tablespace ts_l200
datafile '/dbms/links/line_200' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l201 including contents;
create tablespace ts_l201
datafile '/dbms/links/line_201' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l202 including contents;
create tablespace ts_l202
datafile '/dbms/links/line_202' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l203 including contents;
create tablespace ts_l203
datafile '/dbms/links/line_203' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l204 including contents;
create tablespace ts_l204
datafile '/dbms/links/line_204' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l205 including contents;
create tablespace ts_l205
datafile '/dbms/links/line_205' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l206 including contents;
```

```
create tablespace ts_l206
datafile '/dbms/links/line_206' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l207 including contents;
create tablespace ts_l207
datafile '/dbms/links/line_207' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l208 including contents;
create tablespace ts_l208
datafile '/dbms/links/line_208' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l209 including contents;
create tablespace ts_l209
datafile '/dbms/links/line_209' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l210 including contents;
create tablespace ts_l210
datafile '/dbms/links/line_210' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l211 including contents;
create tablespace ts_l211
datafile '/dbms/links/line_211' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l212 including contents;
create tablespace ts_l212
datafile '/dbms/links/line_212' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l213 including contents;
create tablespace ts_l213
datafile '/dbms/links/line_213' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l214 including contents;
create tablespace ts_l214
datafile '/dbms/links/line_214' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l215 including contents;
create tablespace ts_l215
datafile '/dbms/links/line_215' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l216 including contents;
create tablespace ts_l216
datafile '/dbms/links/line_216' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l217 including contents;
create tablespace ts_l217
datafile '/dbms/links/line_217' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l218 including contents;
create tablespace ts_l218
datafile '/dbms/links/line_218' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l219 including contents;
create tablespace ts_l219
datafile '/dbms/links/line_219' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l220 including contents;
create tablespace ts_l220
datafile '/dbms/links/line_220' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l221 including contents;
create tablespace ts_l221
datafile '/dbms/links/line_221' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l222 including contents;
create tablespace ts_l222
datafile '/dbms/links/line_222' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l223 including contents;
create tablespace ts_l223
datafile '/dbms/links/line_223' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l224 including contents;
create tablespace ts_l224
datafile '/dbms/links/line_224' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l225 including contents;
create tablespace ts_l225
datafile '/dbms/links/line_225' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l226 including contents;
create tablespace ts_l226
datafile '/dbms/links/line_226' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l227 including contents;
create tablespace ts_l227
datafile '/dbms/links/line_227' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l228 including contents;
create tablespace ts_l228
datafile '/dbms/links/line_228' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l229 including contents;
create tablespace ts_l229
datafile '/dbms/links/line_229' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l230 including contents;
create tablespace ts_l230
datafile '/dbms/links/line_230' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l231 including contents;
create tablespace ts_l231
datafile '/dbms/links/line_231' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l232 including contents;
create tablespace ts_l232
datafile '/dbms/links/line_232' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l233 including contents;
create tablespace ts_l233
datafile '/dbms/links/line_233' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l234 including contents;
create tablespace ts_l234
datafile '/dbms/links/line_234' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l235 including contents;
create tablespace ts_l235
datafile '/dbms/links/line_235' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l236 including contents;
create tablespace ts_l236
datafile '/dbms/links/line_236' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l237 including contents;
```

```
create tablespace ts_l237
datafile '/dbms/links/line_237' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l238 including contents;
create tablespace ts_l238
datafile '/dbms/links/line_238' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l239 including contents;
create t ablespace ts_l239
datafile '/dbms/links/line_239' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l240 including contents;
create tablespace ts_l240
datafile '/dbms/links/line_240' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l241 including contents;
create tablespace ts_l241
datafile '/dbms/link s/line_241' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l242 including contents;
create tablespace ts_l242
datafile '/dbms/links/line_242' size 90 00m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l243 including contents;
create tablespace ts_l243
datafile '/dbms/links/line_243' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l244 including contents;
create tablespace ts_l244
datafile '/dbms/links/line_244' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l245 including contents;
create tablespace ts_l245
datafile '/dbms/links/line_245' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l246 including contents;
create tablespace ts_l246
datafile '/dbms/links/line_246' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l247 including contents;
create tablespace ts_l247
datafile '/dbms/links/line_247' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlim ited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l248 including contents;
create tablespace ts_l248
datafile '/dbms/links/line_248' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l249 including contents;
create tablespace ts_l249
datafile '/dbms/links/line_249' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l250 including contents;
create tablespace ts_l250
datafile '/dbms/links/line_250' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l251 including contents;
create tablespace ts_l251
datafile '/dbms/links/line_251' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l252 including contents;
create tablespace ts_l252
datafile '/dbms/links/line_252' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l253 including contents;
create tablespace ts_l253
datafile '/dbms/links/line_253' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l254 including contents;
create tablespace ts_l254
datafile '/dbms/links/line_254' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l255 including contents;
create tablespace ts_l255
datafile '/dbms/links/line_255' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l256 including contents;
create tablespace ts_l256
datafile '/dbms/links/line_256' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l257 including contents;
create tablespace ts_l257
datafile '/dbms/links/line_257' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l258 including contents;
create tablespace ts_l258
datafile '/dbms/links/line_258' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l259 including contents;
create tablespace ts_l259
datafile '/dbms/links/line_259' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l260 including contents;
create tablespace ts_l260
datafile '/dbms/links/line_260' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l261 including contents;
create tablespace ts_l261
datafile '/dbms/links/line_261' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l262 including contents;
create tablespace ts_l262
datafile '/dbms/links/line_262' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l263 including contents;
create tablespace ts_l263
datafile '/dbms/links/line_263' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l264 including contents;
create tablespace ts_l264
datafile '/dbms/links/line_264' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l265 including contents;
create tablespace ts_l265
datafile '/dbms/links/line_265' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l266 including contents;
create tablespace ts_l266
datafile '/dbms/links/line_266' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l267 including contents;
create tablespace ts_l267
datafile '/dbms/links/line_267' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l268 including contents;
```

```
create tablespace ts_l268
datafile '/dbms/links/line_268' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l269 including contents;
create tablespace ts_l269
datafile '/dbms/links/line_269' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l270 including contents;
create tablespace ts_l270
datafile '/dbms/links/line_270' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l271 including contents;
create tablespace ts_l271
datafile '/dbms/links/line_271' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l272 including contents;
create tablespace ts_l272
datafile '/dbms/links/line_272' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l273 including contents;
create tablespace ts_l273
datafile '/dbms/links/line_273' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l274 including contents;
create tablespace ts_l274
datafile '/dbms/links/line_274' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l275 including contents;
create tablespace ts_l275
datafile '/dbms/links/line_275' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l276 including contents;
create tablespace ts_l276
datafile '/dbms/links/line_276' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l277 including contents;
create tablespace ts_l277
datafile '/dbms/links/line_277' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l278 including contents;
create tablespace ts_l278
datafile '/dbms/links/line_278' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l279 including contents;
create tablespace ts_l279
datafile '/dbms/links/line_279' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l280 including contents;
create tablespace ts_l280
datafile '/dbms/links/line_280' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l281 including contents;
create tablespace ts_l281
datafile '/dbms/links/line_281' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l282 including contents;
create tablespace ts_l282
datafile '/dbms/links/line_282' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l283 including contents;
create tablespace ts_l283
datafile '/dbms/links/line_283' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l284 including contents;
create tablespace ts_l284
datafile '/dbms/links/line_284' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l285 including contents;
create tablespace ts_l285
datafile '/dbms/links/line_285' size 9000m reuse
extent management dictionary default st orage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l286 including contents;
create tablespace ts_l286
datafile '/dbms/links/line_286' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l287 including contents;
create tablespace ts_l287
datafile '/dbms/links/line_287' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l288 including contents;
create tablespace ts_l288
datafile '/dbms/links/line_288' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l289 including contents;
create tablespace ts_l289
datafile '/dbms/links/line_289' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l290 including contents;
create tablespace ts_l290
datafile '/dbms/links/line_290' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l291 including contents;
create tablespace ts_l291
datafile '/dbms/links/line_291' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l292 including contents;
create tablespace ts_l292
datafile '/dbms/links/line_292' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l293 including contents;
create tablespace ts_l293
datafile '/dbms/links/line_293' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l294 including contents;
create tablespace ts_l294
datafile '/dbms/links/line_294' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l295 including contents;
create tablespace ts_l295
datafile '/dbms/links/line_295' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l296 including contents;
create tablespace ts_l296
datafile '/dbms/links/line_296' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l297 including contents;
create tablespace ts_l297
datafile '/dbms/links/line_297' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l298 including contents;
create tablespace ts_l298
datafile '/dbms/links/line_298' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l299 including contents;
```

```
create tablespace ts_l299
datafile '/dbms/links/line_299' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l300 including contents;
create tablespace ts_l300
datafile '/dbms/links/line_300' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m  maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l301 including contents;
create tablespace ts_l301
datafile '/dbms/links/line_301' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l302 including contents;
create tablespace ts_l302
datafile '/dbms/links/line_302' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l303 including contents;
create tablespace ts_l303
datafile '/dbms/links/line_303' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l304 including contents;
create tablespace ts_l304
datafile '/dbms/links/line_304' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l305 in cluding contents;
create tablespace ts_l305
datafile '/dbms/links/line_305' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l306 including contents;
create tablespace ts_l306
datafile '/dbms/links/line_306' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l307 including contents;
create tablespace ts_l307
datafile '/dbms/links/line_307' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l308 including contents;
create tablespace ts_l308
datafile '/dbms/links/line_308' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l309 including contents;
create tablespace ts_l309
datafile '/dbms/links/line_309'  size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l310 including contents;
create tablespace ts_l310
datafile '/dbms/links/line_310' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l311 including contents;
create tablespace ts_l311
datafile '/dbms/links/line_311' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l312 including contents;
create tablespace ts_l312
datafile '/dbms/links/line_312' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l313 including contents;
create tablespace ts_l313
datafile '/dbms/links/line_313' size 9000m reuse
extent management dictionary default storage (initial 1050m  next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l314 including contents;
create tablespace ts_l314
datafile '/dbms/links/line_314' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_l315 including contents;
create tablespace ts_l315
datafile '/dbms/links/line_315' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l316 including contents;
create tablespace ts_l316
datafile '/dbms/links/line_316' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l317 including contents;
create tablespace ts_l317
datafile '/dbms/links/line_317' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace t s_l318 including contents;
create tablespace ts_l318
datafile '/dbms/links/line_318' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l319 including contents;
create tablespace ts_l319
datafile '/dbms/links/line_319' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l320 including contents;
create tablespace ts_l320
datafile '/dbms/links/line_320' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l321 including contents;
create tablespace ts_l321
datafile '/dbms/links/line_321' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l322 including contents;
create tablespace ts_l322
datafile '/dbms/links/line_322' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l323 including contents;
create tablespace ts_l323
datafile '/dbms/links/line_323' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l324 including contents;
create tablespace ts_l324
datafile '/dbms/links/line_324' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l325 including contents;
create tablespace ts_l325
datafile '/dbms/links/line_325' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l326 including contents;
create tablespace ts_l326
datafile '/dbms/links/line_326' size 9000m reuse
extent management dictionary default storage (init ial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l327 including contents;
create tablespace ts_l327
datafile '/dbms/links/line_327' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l328 including contents;
create tablespace ts_l328
datafile '/dbms/links/line_328' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l329 including contents;
create tablespace ts_l329
datafile '/dbms/links/line_329' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l330 including contents;
```

```
create tablespace ts_l330
datafile '/dbms/links/line_330' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l331 including contents;
create tablespace ts_l331
datafile '/dbms/links/line_331' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l332 including contents;
create tablespace ts_l332
datafile '/dbms/links/line_332' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l333 including contents;
create tablespace ts_l333
datafile '/dbms/links/line_333' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l334 including contents;
create tablespace ts_l334
datafile '/dbms/links/line_334' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l335 including contents;
create tablespace ts_l335
datafile '/dbms/links/line_335' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_l336 including contents;
create tablespace ts_l336
datafile '/dbms/links/line_336' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
}
*wait
*sql
{
--drop tablespace ts_o1 including contents;
create tablespace ts_o1
datafile '/dbms/links/ord_1' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o2 including contents;
create tablespace ts_o2
datafile '/dbms/links/ord_2' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o3 including contents;
create tablespace ts_o3
datafile '/dbms/links/ord_3' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o4 including contents;
create tablespace ts_o4
datafile '/dbms/links/ord_4' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o5 including contents;
create tablespace ts_o5
datafile '/dbms/links/ord_5' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o6 including contents;
create tablespace ts_o6
datafile '/dbms/links/ord_6' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o7 including contents;
create tablespace ts_o7
datafile '/dbms/links/ord_7' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o8 including contents;
create tablespace ts_o8
datafile '/dbms/links/ord_8' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o9 including contents;
create tablespace ts_o9
datafile '/dbms/links/ord_9' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
```

```
;
}
*sql
{
--drop tablespace ts_o10 including contents;
create tablespace ts_o10
datafile '/dbms/links/ord_10' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o11 including contents;
create tablespace ts_o11
datafile '/dbms/links/ord_11' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o12 including contents;
create tablespace ts_o12
datafile '/dbms/links/ord_12' size 2100m reuse
extent management dictionary default  storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o13 including contents;
create tablespace ts_o13
datafile '/dbms/links/ord_13' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o14 including contents;
create tablespace ts_o14
datafile '/dbms/links/ord_14' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o15 including contents;
create tablespace ts_o15
datafile '/dbms/links/ord_15' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o16 including contents;
create tablespace ts_o16
datafile '/dbms/links/ord_16' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o17 including contents;
create tablespace ts_o17
datafile '/dbms/links/ord_17' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o18 including contents;
create tablespace ts_o18
datafile '/dbms/links/ord_18' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o19 including contents;
create tablespace ts_o19
datafile '/dbms/links/ord_19' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o20 including contents;
create tablespace ts_o20
datafile '/dbms/links/ord_20' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o21 including contents;
create tablespace ts_o21
datafile '/dbms/links/ord_21' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o22 including contents;
create tablespace ts_o22
datafile '/dbms/links/ord_22' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o23 including contents;
create tablespace ts_o23
datafile '/dbms/links/ord_23' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o24 including contents;
create tablespace ts_o24
datafile '/dbms/links/ord_24' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o25 including contents;
```

```
create tablespace ts_o25
datafile '/dbms/links/ord_25' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pct increase 0)
;
}
*sql
{
--drop tablespace ts_o26 including contents;
create tablespace ts_o26
datafile '/dbms/links/ord_26' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o27 including contents;
create tablespace ts_o27
datafile '/dbms/links/ord_27' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o28 including contents;
create tablespace ts_o28
datafile '/dbms/links/ord_28' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o29 including contents;
create tablespace ts_o29
datafile '/dbms/links/ord_29' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o30 including contents;
create tablespace ts_o30
datafile '/dbms/links/ord_30' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o31 including contents;
create tablespace ts_o31
datafile '/dbms/links/ord_31' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o32 including contents;
create tablespace ts_o32
datafile '/dbms/links/ord_32' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o33 including contents;
create tablespace ts_o33
datafile '/dbms/links/ord_33' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o34 including contents;
create tablespace ts_o34
datafile '/dbms/links/ord_34' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o35 including contents;
create tablespace ts_o35
datafile '/dbms/links/ord_35' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o36 including contents;
create tablespace ts_o36
datafile '/dbms/links/ord_36' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o37 including contents;
create tablespace ts_o37
datafile '/dbms/links/ord_37' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o38 including contents;
create tablespace ts_o38
datafile '/dbms/links/ord_38' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o39 including contents;
create tablespace ts_o39
datafile '/dbms/links/ord_39' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o40 including contents;
create tablespace ts_o40
datafile '/dbms/links/ord_40' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o41 including contents;
create tablespace ts_o41
datafile '/dbms/links/ord_41' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o42 including contents;
create tablespace ts_o42
datafile '/dbms/links/ord_42' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o43 including contents;
create tablespace ts_o43
datafile '/dbms/links/ord_43' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o44 including contents;
create tablespace ts_o44
datafile '/dbms/links/ord_44' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o45 including contents;
create tablespace ts_o45
datafile '/dbms/links/ord_45' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o46 including contents;
create tablespace ts_o46
datafile '/dbms/links/ord_46' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o47 including contents;
create tablespace ts_o47
datafile '/dbms/links/ord_47' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o48 including contents;
create tablespace ts_o48
datafile '/dbms/links/ord_48' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o49 including contents;
create tablespace ts_o49
datafile '/dbms/links/ord_49' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o50 including contents;
create tablespace ts_o50
datafile '/dbms/links/ord_50' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o51 including contents;
create tablespace ts_o51
datafile '/dbms/links/ord_51' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o52 including contents;
create tablespace ts_o52
datafile '/dbms/links/ord_52' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o53 including contents;
create tablespace ts_o53
datafile '/dbms/links/ord_53' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o54 including contents;
create tablespace ts_o54
datafile '/dbms/links/ord_54' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o55 including contents;
create tablespace ts_o55
datafile '/dbms/links/ord_55' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o56 including contents;
```

```
create tablespace ts_o56
datafile '/dbms/links/ord_56' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o57 including contents;
create tablespace ts_o57
datafile '/dbms/links/ord_57' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pct increase 0)
;
}
*sql
{
--drop tablespace ts_o58 including contents;
create tablespace ts_o58
datafile '/dbms/links/ord_58' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o59 including contents;
create tablespace ts_o59
datafile '/dbms/links/ord_59' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o60 including contents;
create tablespace ts_o60
datafile '/dbms/links/ord_60' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o61 including contents;
create tablespace ts_o61
datafile '/dbms/links/ord_61' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o62 including contents;
create tablespace ts_o62
datafile '/dbms/links/ord_62' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o63 including contents;
create tablespace ts_o63
datafile '/dbms/links/ord_63' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o64 including contents;
create tablespace ts_o64
datafile '/dbms/links/ord_64' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o65 including contents;
create tablespace ts_o65
datafile '/dbms/links/ord_65' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o66 including contents;
create tablespace ts_o66
datafile '/dbms/links/ord_66' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o67 including contents;
create tablespace ts_o67
datafile '/dbms/links/ord_67' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o68 including contents;
create tablespace ts_o68
datafile '/dbms/links/ord_68' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o69 including contents;
create tablespace ts_o69
datafile '/dbms/links/ord_69' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o70 including contents;
create tablespace ts_o70
datafile '/dbms/links/ord_70' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o71 including contents;
create tablespace ts_o71
datafile '/dbms/links/ord_71' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o72 including contents;
create tablespace ts_o72
datafile '/dbms/links/ord_72' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o73 including contents;
create tablespace ts_o73
datafile '/dbms/links/ord_73' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o74 including contents;
create tablespace ts_o74
datafile '/dbms/links/ord_74' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o75 including contents;
create tablespace ts_o75
datafile '/dbms/links/ord_75' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o76 including contents;
create tablespace ts_o76
datafile '/dbms/links/ord_76' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o77 including contents;
create tablespace ts_o77
datafile '/dbms/links/ord_77' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o78 including contents;
create tablespace ts_o78
datafile '/dbms/links/ord_78' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o79 including contents;
create tablespace ts_o79
datafile '/dbms/links/ord_79' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o80 including contents;
create tablespace ts_o80
datafile '/dbms/links/ord_80' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o81 including contents;
create tablespace ts_o81
datafile '/dbms/links/ord_81' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o82 including contents;
create tablespace ts_o82
datafile '/dbms/links/ord_82' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o83 including contents;
create tablespace ts_o83
datafile '/dbms/links/ord_83' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o84 including contents;
create tablespace ts_o84
datafile '/dbms/links/ord_84' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o85 including contents;
create tablespace ts_o85
datafile '/dbms/links/ord_85' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o86 including contents;
create tablespace ts_o86
datafile '/dbms/links/ord_86' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o87 including contents;
```

```
create tablespace ts_o87
datafile '/dbms/links/ord_87' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o88 including contents;
create tablespace ts_o88
datafile '/dbms/links/ord_88' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o89 including contents;
create tablespace ts_o89
datafile '/dbms/links/ord_89' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pct increase 0)
;
}
*sql
{
--drop tablespace ts_o90 including contents;
create tablespace ts_o90
datafile '/dbms/links/ord_90' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o91 including contents;
create tablespace ts_o91
datafile '/dbms/links/ord_91' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o92 including contents;
create tablespace ts_o92
datafile '/dbms/links/ord_92' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o93 including contents;
create tablespace ts_o93
datafile '/dbms/links/ord_93' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o94 including contents;
create tablespace ts_o94
datafile '/dbms/links/ord_94' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o95 including contents;
create tablespace ts_o95
datafile '/dbms/links/ord_95' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o96 including contents;
create tablespace ts_o96
datafile '/dbms/links/ord_96' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o97 including contents;
create tablespace ts_o97
datafile '/dbms/links/ord_97' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o98 including contents;
create tablespace ts_o98
datafile '/dbms/links/ord_98' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o99 including contents;
create tablespace ts_o99
datafile '/dbms/links/ord_99' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o100 including contents;
create tablespace ts_o100
datafile '/dbms/links/ord_100' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o101 including contents;
create tablespace ts_o101
datafile '/dbms/links/ord_101' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o102 including contents;
create tablespace ts_o102
datafile '/dbms/links/ord_102' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o103 including contents;
create tablespace ts_o103
datafile '/dbms/links/ord_103' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o104 including contents;
create tablespace ts_o104
datafile '/dbms/links/ord_104' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o105 including contents;
create tablespace ts_o105
datafile '/dbms/links/ord_105' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o106 including contents;
create tablespace ts_o106
datafile '/dbms/links/ord_106' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o107 including contents;
create tablespace ts_o107
datafile '/dbms/links/ord_107' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o108 including contents;
create tablespace ts_o108
datafile '/dbms/links/ord_108' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o109 including contents;
create tablespace ts_o109
datafile '/dbms/links/ord_109' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o110 including contents;
create tablespace ts_o110
datafile '/dbms/links/ord_110' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o111 including contents;
create tablespace ts_o111
datafile '/dbms/links/ord_111' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o112 including contents;
create tablespace ts_o112
datafile '/dbms/links/ord_112' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o113 including contents;
create tablespace ts_o113
datafile '/dbms/links/ord_113' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o114 including contents;
create tablespace ts_o114
datafile '/dbms/links/ord_114' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o115 including contents;
create tablespace ts_o115
datafile '/dbms/links/ord_115' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o116 including contents;
create tablespace ts_o116
datafile '/dbms/links/ord_116' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o117 including contents;
create tablespace ts_o117
datafile '/dbms/links/ord_117' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o118 including contents;
```

```
create tablespace ts_o118
datafile '/dbms/links/ord_118' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o119 including contents;
create tablespace ts_o119
datafile '/dbms/links/ord_119' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o120 including contents;
create tablespace ts_o120
datafile '/dbms/links/ord_120' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o121 including contents;
create tablespace ts_o121
datafile '/dbms/links/ord_121' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o122 including contents;
create tablespace ts_o122
datafile '/dbms/links/ord_122' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o123 including contents;
create tablespace ts_o123
datafile '/dbms/links/ord_123' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o124 including contents;
create tablespace ts_o124
datafile '/dbms/links/ord_124' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o125 including contents;
create tablespace ts_o125
datafile '/dbms/links/ord_125' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o126 including contents;
create tablespace ts_o126
datafile '/dbms/links/ord_126' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o127 including contents;
create tablespace ts_o127
datafile '/dbms/links/ord_127' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o128 including contents;
create tablespace ts_o128
datafile '/dbms/links/ord_128' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o129 including contents;
create tablespace ts_o129
datafile '/dbms/links/ord_129' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o130 including contents;
create tablespace ts_o130
datafile '/dbms/links/ord_130' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o131 including contents;
create tablespace ts_o131
datafile '/dbms/links/ord_131' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o132 including contents;
create tablespace ts_o132
datafile '/dbms/links/ord_132' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o133 including contents;
create tablespace ts_o133
datafile '/dbms/links/ord_133' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o134 including contents;
create tablespace ts_o134
datafile '/dbms/links/ord_134' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o135 including contents;
create tablespace ts_o135
datafile '/dbms/links/ord_135' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o136 including contents;
create tablespace ts_o136
datafile '/dbms/links/ord_136' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o137 including contents;
create tablespace ts_o137
datafile '/dbms/links/ord_137' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o138 including contents;
create tablespace ts_o138
datafile '/dbms/links/ord_138' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o139 including contents;
create tablespace ts_o139
datafile '/dbms/links/ord_139' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o140 including contents;
create tablespace ts_o140
datafile '/dbms/links/ord_140' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o141 including contents;
create tablespace ts_o141
datafile '/dbms/links/ord_141' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o142 including contents;
create tablespace ts_o142
datafile '/dbms/links/ord_142' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o143 including contents;
create tablespace ts_o143
datafile '/dbms/links/ord_143' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o144 including contents;
create tablespace ts_o144
datafile '/dbms/links/ord_144' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o145 including contents;
create tablespace ts_o145
datafile '/dbms/links/ord_145' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o146 including contents;
create tablespace ts_o146
datafile '/dbms/links/ord_146' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o147 including contents;
create tablespace ts_o147
datafile '/dbms/links/ord_147' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o148 including contents;
create tablespace ts_o148
datafile '/dbms/links/ord_148' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o149 including contents;
```

```
create tablespace ts_o149
datafile '/dbms/links/ord_149' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o150 including contents;
create tablespace ts_o150
datafile '/dbms/links/ord_150' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop t ablespace ts_o151 including contents;
create tablespace ts_o151
datafile '/dbms/links/ord_151' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o152 including contents;
create tablespace ts_o152
datafile '/dbms/links/ord_152' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o153 including contents;
create tablespace ts_o153
datafile '/dbms/links/ord_153' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o154 including contents;
create tablespace ts_o154
datafile '/dbms/links/ord_154' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o155 including contents;
create tablespace ts_o155
datafile '/dbms/links/ord_155' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o156 including contents;
create tablespace ts_o156
datafile '/dbms/links/ord_156' size 2100m  reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o157 including contents;
create tablespace ts_o157
datafile '/dbms/links/ord_157' size 2100m reuse
extent managem ent dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o158 including contents;
create tablespace ts_o158
datafile '/dbms/links/ord_158' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o159 including contents;
create tablespace ts_o159
datafile '/dbms/links/ord_159' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o160 including contents;
create tablespace ts_o160
datafile '/dbms/links/ord_160' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o161 including contents;
create tablespace ts_o161
datafile '/dbms/links/ord_161' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o162 including contents;
create tablespace ts_o162
datafile '/dbms/links/ord_162' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o163 including contents;
create tablespace ts_o163
datafile '/dbms/links/ord_163' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o164 including contents;
create tablespace ts_o164
datafile '/dbms/links/ord_164' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o165 including contents;
create tablespace ts_o165
datafile '/dbms/links/ord_165' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o166 including contents;
create tablespace ts_o166
datafile '/dbms/links/ord_166' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o167 including contents;
create tablespace ts_o167
datafile '/dbms/links/ord_167' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o168 including contents;
create tablespace ts_o168
datafile '/dbms/links/ord_168' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o169 including contents;
create tablespace ts_o169
datafile '/dbms/links/ord_169' size 2100m reuse
extent man agement dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o170 including contents;
create tablespace ts_o170
datafile '/dbms/links/ord_170' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o171 including contents;
create tablespace ts_o171
datafile '/dbms/links/ord_171' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o172 including contents;
create tablespace ts_o172
datafile '/dbms/links/ord_172' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o173 including contents;
create tablespace ts_o173
datafile '/dbms/links/ord_173' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctin crease 0)
;
}
*sql
{
--drop tablespace ts_o174 including contents;
create tablespace ts_o174
datafile '/dbms/links/ord_174' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o175 including contents;
create tablespace ts_o175
datafile '/dbms/links/ord_175' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o176 including contents;
create tablespace ts_o176
datafile '/dbms/links/ord_176' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o177 including contents;
create tablespace ts_o177
datafile '/dbms/links/ord_177' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o178 including contents;
create tablespace ts_o178
datafile '/dbms/links/ord_178' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o179 including contents;
create tablespace ts_o179
datafile '/dbms/links/ord_179' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o180 including contents;
```

```
create tablespace ts_o180
datafile '/dbms/links/ord_180' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o181 including contents;
create tablespace ts_o181
datafile '/dbms/links/ord_181' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o182 including contents;
create tablespace ts_o182
datafile '/dbms/links/ord_182' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o183 including contents;
create tablespace ts_o183
datafile '/dbms/links/ord_183' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o184 including contents;
create tablespace ts_o184
datafile '/dbms/links/ord_184' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o185 including contents;
create tablespace ts_o185
datafile '/dbms/links/ord_185' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o186 including contents;
create tablespace ts_o186
datafile '/dbms/links/ord_186' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o187 including contents;
create tablespace ts_o187
datafile '/dbms/links/ord_187' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o188 including contents;
create tablespace ts_o188
datafile '/dbms/links/ord_188' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o189 including contents;
create tablespace ts_o189
datafile '/dbms/links/ord_189' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o190 including contents;
create tablespace ts_o190
datafile '/dbms/links/ord_190' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o191 including contents;
create tablespace ts_o191
datafile '/dbms/links/ord_191' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o192 including contents;
create tablespace ts_o192
datafile '/dbms/links/ord_192' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o193 including contents;
create tablespace ts_o193
datafile '/dbms/links/ord_193' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o194 including contents;
create tablespace ts_o194
datafile '/dbms/links/ord_194' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o195 including contents;
create tablespace ts_o195
datafile '/dbms/links/ord_195' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o196 including contents;
create tablespace ts_o196
datafile '/dbms/links/ord_196' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o197 including contents;
create tablespace ts_o197
datafile '/dbms/links/ord_197' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimit ed pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o198 including contents;
create tablespace ts_o198
datafile '/dbms/links/ord_198' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o199 including contents;
create tablespace ts_o199
datafile '/dbms/links/ord_199' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o200 including contents;
create tablespace ts_o200
datafile '/dbms/links/ord_200' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o201 includin g contents;
create tablespace ts_o201
datafile '/dbms/links/ord_201' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o202 including contents;
create tablespace ts_o202
datafile '/dbms/links/ord_202' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o203 including contents;
create tablespace ts_o203
datafile '/dbms/links/ord_203' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o204 including contents;
create tablespace ts_o204
datafile '/dbms/links/ord_204' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o205 including contents;
create tablespace ts_o205
datafile '/dbms/links/ord_205' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o206 including contents;
create tablespace ts_o206
datafile '/dbms/links/ord_206' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o207 including contents;
create tablespace ts_o207
datafile '/dbms/links/ord_207' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o208 including contents;
create tablespace ts_o208
datafile '/dbms/links/ord_208' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o209 including contents;
create tablespace ts_o209
datafile '/dbms/links/ord_209' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o210 including contents;
create tablespace ts_o210
datafile '/dbms/links/ord_210' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o211 including contents;
```

```
create tablespace ts_o211
datafile '/dbms/links/ord_211' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o212 including contents;
create tablespace ts_o212
datafile '/dbms/links/ord_212' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o213 including contents;
create tablespace ts_o213
datafile '/dbms/links/ord_213' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o214 including contents;
create tablespace ts_o214
datafile '/dbms/links/ord_214' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o215 including contents;
create tablespace ts_o215
datafile '/dbms/links/ord_215' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o216 including contents;
create tablespace ts_o216
datafile '/dbms/links/ord_216' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o217 including contents;
create tablespace ts_o217
datafile '/dbms/links/ord_217' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o218 including contents;
create tablespace ts_o218
datafile '/dbms/links/ord_218' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o219 including contents;
create tablespace ts_o219
datafile '/dbms/links/ord_219' size 2100m reuse
extent management dictionary default  storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o220 including contents;
create tablespace ts_o220
datafile '/dbms/links/ord_220' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o221 including contents;
create tablespace ts_o221
datafile '/dbms/links/ord_221' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o222 including contents;
create tablespace ts_o222
datafile '/dbms/links/ord_222' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o223 including contents;
create tablespace ts_o223
datafile '/dbms/links/ord_223' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o224 including contents;
create tablespace ts_o224
datafile '/dbms/links/ord_224' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o225 including contents;
create tablespace ts_o225
datafile '/dbms/links/ord_225' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o226 including contents;
create tablespace ts_o226
datafile '/dbms/links/ord_226' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o227 including contents;
create tablespace ts_o227
datafile '/dbms/links/ord_227' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o228 including contents;
create tablespace ts_o228
datafile '/dbms/links/ord_228' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o229 including contents;
create tablespace ts_o229
datafile '/dbms/links/ord_229' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o230 including contents;
create tablespace ts_o230
datafile '/dbms/links/ord_230' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o231 including contents;
create tablespace ts_o231
datafile '/dbms/links/ord_231' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o232 including contents;
create tablespace ts_o232
datafile '/dbms/links/ord_232' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o233 including contents;
create tablespace ts_o233
datafile '/dbms/links/ord_233' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o234 including contents;
create tablespace ts_o234
datafile '/dbms/links/ord_234' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o235 including contents;
create tablespace ts_o235
datafile '/dbms/links/ord_235' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o236 including contents;
create tablespace ts_o236
datafile '/dbms/links/ord_236' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o237 including contents;
create tablespace ts_o237
datafile '/dbms/links/ord_237' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o238 including contents;
create tablespace ts_o238
datafile '/dbms/links/ord_238' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o239 including contents;
create tablespace ts_o239
datafile '/dbms/links/ord_239' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o240 including contents;
create tablespace ts_o240
datafile '/dbms/links/ord_240' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o241 including contents;
create tablespace ts_o241
datafile '/dbms/links/ord_241' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o242 including contents;
```

```
create tablespace ts_o242
datafile '/dbms/links/ord_242' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o243 including contents;
create tablespace ts_o243
datafile '/dbms/links/ord_243' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o244 including contents;
create tablespace ts_o244
datafile '/dbms/links/ord_244' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o245 including contents;
create tablespace ts_o245
datafile '/dbms/links/ord_245' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o246 including contents;
create tablespace ts_o246
datafile '/dbms/links/ord_246' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o247 including contents;
create tablespace ts_o247
datafile '/dbms/links/ord_247' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o248 including contents;
create tablespace ts_o248
datafile '/dbms/links/ord_248' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o249 including contents;
create tablespace ts_o249
datafile '/dbms/links/ord_249' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o250 including contents;
create tablespace ts_o250
datafile '/dbms/links/ord_250' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o251 including contents;
create tablesp ace ts_o251
datafile '/dbms/links/ord_251' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o252 including contents;
create tablespace ts_o252
datafile '/dbms/links/ord_252' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o253 including contents;
create tablespace ts_o253
datafile '/dbms/links/ord_253' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o254 including contents;
create tablespace ts_o254
datafile '/dbms/links/ord_254' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o255 including contents;
create tablespace ts_o255
datafile '/dbms/links/ord_255' size 2100m reuse
extent management dictio nary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o256 including contents;
create tablespace ts_o256
datafile '/dbms/links/ord_256' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o257 including contents;
create tablespace ts_o257
datafile '/dbms/links/ord_257' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o258 including contents;
create tablespace ts_o258
datafile '/dbms/links/ord_258' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o259 including contents;
create tablespace ts_o259
datafile '/dbms/links/ord_259' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o260 including contents;
create tablespace ts_o260
datafile '/dbms/links/ord_260' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o261 including contents;
create tablespace ts_o261
datafile '/dbms/links/ord_261' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o262 including contents;
create tablespace ts_o262
datafile '/dbms/links/ord_262' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o263 including contents;
create tablespace ts_o263
datafile '/dbms/links/ord_263' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o264 including contents;
create tablespace ts_o264
datafile '/dbms/links/ord_264' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o265 including contents;
create tablespace ts_o265
datafile '/dbms/links/ord_265' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o266 including contents;
create tablespace ts_o266
datafile '/dbms/links/ord_266' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o267 including contents;
create tablespace ts_o267
datafile '/dbms/links/ord_267' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o268 including contents;
create tablespace ts_o268
datafile '/dbms/links/ord_268' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o269 including contents;
create tablespace ts_o269
datafile '/dbms/links/ord_269' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o270 including contents;
create tablespace ts_o270
datafile '/dbms/links/ord_270' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o271 including contents;
create tablespace ts_o271
datafile '/dbms/links/ord_271' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o272 including contents;
create tablespace ts_o272
datafile '/dbms/links/ord_272' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o273 including contents;
```

```
create tablespace ts_o273
datafile '/dbms/links/ord_273' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o274 including contents;
create tablespace ts_o274
datafile '/dbms/links/ord_274' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o275 including contents;
create tablespace ts_o275
datafile '/dbms/links/ord_275' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o276 including contents;
create tablespace ts_o276
datafile '/dbms/links/ord_276' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o277 including contents;
create tablespace ts_o277
datafile '/dbms/links/ord_277' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o278 including contents;
create tablespace ts_o278
datafile '/dbms/links/ord_278' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o279 including contents;
create tablespace ts_o279
datafile '/dbms/links/ord_279' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o280 including contents;
create tablespace ts_o280
datafile '/dbms/links/ord_280' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o281 including contents;
create tablespace ts_o281
datafile '/dbms/links/ord_281' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o282 including contents;
create tablespace ts_o282
datafile '/dbms/links/ord_282' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o283 including contents;
create tablespace ts_o283
datafile '/dbms/links/ord_283' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o284 including contents;
create tablespace ts_o284
datafile '/dbms/links/ord_284' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o285 including contents;
create tablespace ts_o285
datafile '/dbms/links/ord_285' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o286 including contents;
create tablespace ts_o286
datafile '/dbms/links/ord_286' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o287 including contents;
create tablespace ts_o287
datafile '/dbms/links/ord_287' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o288 including contents;
create tablespace ts_o288
datafile '/dbms/links/ord_288' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o289 including contents;
create tablespace ts_o289
datafile '/dbms/links/ord_289' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o290 including contents;
create tablespace ts_o290
datafile '/dbms/links/ord_290' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o291 including contents;
create tablespace ts_o291
datafile '/dbms/links/ord_291' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o292 including contents;
create tablespace ts_o292
datafile '/dbms/links/ord_292' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o293 including contents;
create tablespace ts_o293
datafile '/dbms/links/ord_293' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o294 including contents;
create tablespace ts_o294
datafile '/dbms/links/ord_294' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o295 including contents;
create tablespace ts_o295
datafile '/dbms/links/ord_295' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o296 including contents;
create tablespace ts_o296
datafile '/dbms/links/ord_296' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o297 including contents;
create tablespace ts_o297
datafile '/dbms/links/ord_297' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o298 including contents;
create tablespace ts_o298
datafile '/dbms/links/ord_298' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o299 including contents;
create tablespace ts_o299
datafile '/dbms/links/ord_299' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o300 including contents;
create tablespace ts_o300
datafile '/dbms/links/ord_300' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o301 including contents;
create tablespace ts_o301
datafile '/dbms/links/ord_301' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o302 including contents;
create tablespace ts_o302
datafile '/dbms/links/ord_302' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o303 including contents;
create tablespace ts_o303
datafile '/dbms/links/ord_303' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o304 including contents;
```

```
create tablespace ts_o304
datafile '/dbms/links/ord_304' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o305 including contents;
create tablespace ts_o305
datafile '/dbms/links/ord_305' size 2100m reuse
extent management dictionary default storage (init ial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o306 including contents;
create tablespace ts_o306
datafile '/dbms/links/ord_306' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
max extents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o307 including contents;
create tablespace ts_o307
datafile '/dbms/links/ord_307' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pct increase 0)
;
}
*sql
{
--drop tablespace ts_o308 including contents;
create tablespace ts_o308
datafile '/dbms/links/ord_308' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o309 including contents;
create tablespace ts_o309
datafile '/dbms/links/ord_309' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace t s_o310 including contents;
create tablespace ts_o310
datafile '/dbms/links/ord_310' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o311 including cont ents;
create tablespace ts_o311
datafile '/dbms/links/ord_311' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o312 including contents;
create tablespace ts_o312
datafile '/dbms/links/ord_312' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o313 including contents;
create tablespace ts_o313
datafile '/dbms/links/ord_313' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o314 including contents;
create tablespace ts_o314
datafile '/dbms/links/ord_314' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o315 including contents;
create tablespace ts_o315
datafile '/dbms/links/ord_315' size 2100m reuse
ext ent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o316 including contents;
create tablespace ts_o316
datafile '/dbms/links/ord_316' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o317 including contents;
create tablespace ts_o317
datafile '/dbms/links/ord_317' size 2100m reuse
extent management dictionary default storage ( initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o318 including contents;
create tablespace ts_o318
datafile '/dbms/links/ord_318' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o319 including contents;
create tablespace ts_o319
datafile '/dbms/links/ord_319' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
```

```
}
*sql
{
--drop tablespace ts_o320 including contents;
create tablespace ts_o320
datafile '/dbms/links/ord_320' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o321 including contents;
create tablespace ts_o321
datafile '/dbms/links/ord_321' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o322 including contents;
create tablespace ts_o322
datafile '/dbms/links/ord_322' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o323 including contents;
create tablespace ts_o323
datafile '/dbms/links/ord_323' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o324 including contents;
create tablespace ts_o324
datafile '/dbms/links/ord_324' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o325 including contents;
create tablespace ts_o325
datafile '/dbms/links/ord_325' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o326 including contents;
create tablespace ts_o326
datafile '/dbms/links/ord_326' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o327 including contents;
create tablespace ts_o327
datafile '/dbms/links/ord_327' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o328 including contents;
create tablespace ts_o328
datafile '/dbms/links/ord_328' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o329 including contents;
create tablespace ts_o329
datafile '/dbms/links/ord_329' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o330 including contents;
create tablespace ts_o330
datafile '/dbms/links/ord_330' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o331 including contents;
create tablespace ts_o331
datafile '/dbms/links/ord_331' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlim ited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o332 including contents;
create tablespace ts_o332
datafile '/dbms/links/ord_332' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o333 including contents;
create tablespace ts_o333
datafile '/dbms/links/ord_333' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o334 including contents;
create tablespace ts_o334
datafile '/dbms/links/ord_334' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o335 including contents;
```

```
create tablespace ts_o335
datafile '/dbms/links/ord_335' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_o336 including contents;
create tablespace ts_o336
datafile '/dbms/links/ord_336' size 2100m reuse
extent management dictionary default storage (initial 450m next 10m
maxextents unlimited pctincrease 0)
;
}
*wait
*sql
{
--drop tablespace ts_small1 including contents;
create tablespace ts_small1
datafile '/dbms/links/small_1' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small2 including contents;
create tablespace ts_small2
datafile '/dbms/links/small_2' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small3 including contents;
create tablespace ts_small3
datafile '/dbms/links/small_3' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small4 including contents;
create tablespace ts_small4
datafile '/dbms/links/small_4' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small5 including contents;
create tablespace ts_small5
datafile '/dbms/links/small_5' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small6 including contents;
create tablespace ts_small6
datafile '/dbms/links/small_6' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small7 including contents;
create tablespace ts_small7
datafile '/dbms/links/small_7' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small8 including contents;
create tablespace ts_small8
datafile '/dbms/links/small_8' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small9 including contents;
create tablespace ts_small9
datafile '/dbms/links/small_9' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small10 including contents;
create tablespace ts_small10
datafile '/dbms/links/small_10' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small11 including contents;
create tablespace ts_small11
datafile '/dbms/links/small_11' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small12 including contents;
create tablespace ts_small12
datafile '/dbms/links/small_12' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small13 including contents;
create tablespace ts_small13
datafile '/dbms/links/small_13' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small14 including contents;
create tablespace ts_small14
datafile '/dbms/links/small_14' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
```

```
;
}
*sql
{
--drop tablespace ts_small15 including contents;
create tablespace ts_small15
datafile '/dbms/links/small_15' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_small16 including contents;
create tablespace ts_small16
datafile '/dbms/links/small_16' size 12280m reuse
extent management dictionary default storage (initial 450m next 20m
maxextents unlimited pctincrease 0)
;
}
*sql
{
--drop tablespace ts_okey including contents;
create tablespace ts_okey
datafile '/dbms/links/okey_1' size 15100m reuse
extent management local
autoallocate
;
}
*sql
{
--drop tablespace ts_custkey including contents;
create tablespace ts_custkey
datafile '/dbms/links/custkey_1' size 1023m reuse
extent management local
uniform size 30M
;
}
*sql
{
--drop tablespace ts_lokey including contents;
create tablespace ts_lokey
datafile '/dbms/links/lokey_1' size 15350m reuse
extent management local
autoallocate
;
}
*sql
{
--drop tablespace ts_psupp1 including contents;
create tablespace ts_psupp1
datafile '/dbms/links/psupp_1' size 14330m reuse
extent management dictionary default storage (initial 1100m next
50m maxextents unlimited pctincrease 0)
;
}
# creating tpch's ts_temp tablespace
*sql
{
--drop tablespace ts_temp including contents;
create temporary tablespace ts_temp
tempfile '/dbms/links/temp_1' size 32000m reuse
extent management local
uniform size 5M
;
}
*wait
# adding tablespace datafiles
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_2' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_3' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_4' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_5' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_6' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_7' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_8' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_9' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_10' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_11' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_12' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_13' size 14330m reuse
;
}
```

```
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_14' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_15' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_16' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_17' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_18' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_19' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_20' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_21' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_22' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_23' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_24' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_25' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_26' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_27' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_28' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_29' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_30' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_31' size 14330m reuse
;
}
*sql
{
alter tablespace ts_psupp1
add datafile '/dbms/links/psupp_32' size 14330m reuse
;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_2' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_3' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_4' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_5' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_6' size 15100m reuse;
}
*sql
```

```
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_7' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_8' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_9' size 15100m reuse;

}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_10' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_11' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_12' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_13' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_14' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_15' size 15100m reuse;
}
*sql
{
alter tablespace ts_okey
  add datafile '/dbms/links/okey_16' size 15100m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_2' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_3' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_4' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_5' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_6' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_7' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_8' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_9' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_10' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_11' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_12' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_13' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_14' size 1023m reuse;
}
*sql
{
alter tablespace t s_custkey
  add datafile '/dbms/links/custkey_15' size 1023m reuse;
}
*sql
{
alter tablespace ts_custkey
  add datafile '/dbms/links/custkey_16' size 1023m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_2' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_3' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_4' size 15350m reuse;
}
```

```
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_5' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_6' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_7' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_8' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_9' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_10' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_11' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_12' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_13' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_14' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_15' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_16' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_17' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_18' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_19' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_20' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_21' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_22' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_23' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_24' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_25' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_26' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_27' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_28' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_29' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_30' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_31' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_32' size 15350m reuse;
}
```

```
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_33' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_34' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_35' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_36' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_37' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_38' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_39' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_40' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_41' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_42' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_43' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_44' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_45' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_46' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_47' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_48' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_49' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_50' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_51' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_52' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_53' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_54' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_55' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_56' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_57' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_58' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_59' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_60' size 15350m reuse;
}
```

```
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_61' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_62' size 15350m reuse;
}
*sql
{
alter tablespace ts_lokey
  add datafile '/dbms/links/lokey_63' size 15350m reuse;
}
*sql
{
alter tablespace ts_ lokey
  add datafile '/dbms/links/lokey_64' size 15350m reuse;
}
*wait
#adding tpch's ts_temp add datafiles
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_2' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_3' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_4' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_5' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_6' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_7' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_8' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_9' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_10' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_11' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_12' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_13' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_14' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_15' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_16' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_17' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_18' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_19' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_20' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_21' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_22' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_23' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_24' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
```

```
add tempfile '/dbms/links/temp_25' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_26' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_27' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_28' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_29' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_30' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_31' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_32' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_33' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_34' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_35' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_36' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_37' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_38' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_39' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_40' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_41' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_42' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_43' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_44' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_45' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_46' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_47' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_48' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_49' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_50' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_51' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_52' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
```

```
add tempfile '/dbms/links/temp_53' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_54' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_55' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_56' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_57' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_58' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_59' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_60' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_61' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_62' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_63' size 32000m reuse;
}
*sql
{
alter tablespace ts_temp
add tempfile '/dbms/links/temp_64' size 32000m reuse;
}
*wait
#adding tpch's ts_undo add datafiles
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_2' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_3' size 22000m reuse;
}
*sql
{
alter tablespace t s_undo
 add datafile '/dbms/links/undo_4' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_5' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_6' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_7' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_8' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_9' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_10' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_11' size 22000m reuse;
}
*sql
{
alter tablespace ts_undo
 add datafile '/dbms/links/undo_12' size 22000m reuse;
}
*wait
*bgoff
%e-sctso
%b-dapop
*bgon=1
#########################################################
#####################
# Schema Creation Phase - User and Tables
#  AND  Database Population Phase
#
# creating tpch user
#*sql
#{
#shutdown abort;
#}
#*wait
#*sql
#{
#startup pfile=/oracle/dbs/3TB_run.ora
#}
#*wait
*sql
{
#drop user tpch cascade;
grant DBA
to tpch identified by tpch;
```

```
}
*wait
*sql
{
connect tpch/tpch;
drop directory data_dir;
create directory data_dir as '/dbms/flat';
}
*sql
{
connect tpch/tpch;
drop table l_et;
create table l_et(
   l_orderkey          number ,
   l_partkey           number ,
   l_suppkey           number ,
   l_linenumber        number ,
   l_quantity          number ,
   l_extendedprice     number ,
   l_discount          number ,
   l_tax               number ,
   l_returnflag        char(1) ,
   l_linestatus        char(1) ,
   l_shipdate          date ,
   l_commitdate        date ,
   l_receiptdate       date ,
   l_shipinstruct      char(25) ,
   l_shipmode          char(10) ,
   l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        badfile 'l_et.bad'
                        logfile 'l_et.log'
                        fields terminated by '|'
                        missing field values are null
            )
            location (
      'lineitem.tbl.1',
      'lineitem.tbl.2',
      'lineitem.tbl.3',
      'lineitem.tbl.4',
      'lineitem.tbl.5',
      'lineitem.tbl.6',
      'lineitem.tbl.7',
      'lineitem.tbl.8',
      'lineitem.tbl.9',
      'lineitem.tbl.10',
      'lineitem.tbl.11',
      'lineitem.tbl.12',
      'lineitem.tbl.13',
      'lineitem.tbl.14',
      'lineitem.tbl.15',
      'lineitem.tbl.16',
      'lineitem.tbl.17',
      'lineitem.tbl.18',
      'lineitem.tbl.19',
      'lineitem.tbl.20',
      'lineitem.tbl.21',
      'lineitem.tbl.22',
      'lineitem.tbl.23',
      'lineitem.tbl.24',
      'lineitem.tbl.25',
      'lineitem.tbl.26',
      'lineitem.t bl.27',
      'lineitem.tbl.28',
      'lineitem.tbl.29',
      'lineitem.tbl.30',
      'lineitem.tbl.31',
      'lineitem.tbl.32',
      'lineitem.tbl.33',
      'lineitem.tbl.34',
      'lineitem.tbl.35',
      'lineitem.tbl.36',
      'lineitem.tbl.37',
      'lineitem.tbl.38',
      'lineitem.tbl.39',
      'lineitem.tbl.40',
      'lineitem.tbl.41',
      'lineitem.tbl.42',
      'lineitem.tbl.43',
      'lineitem.tbl.44',
      'lineitem.tbl.45',
      'lineitem.tbl.46',
      'lineitem.tbl.47',
      'lineitem.tbl.48',
      'lineitem.tbl.49',
      'lineitem.tbl.50',
      'lineitem.tbl.51',
      'lineitem.tbl.52',
      'lineitem.tbl.53',
      'lineitem.tbl.54',
      'lineitem.tbl.55',
      'lineitem.tbl.56',
      'lineitem.tbl.57',
      'lineitem.tbl.58',
      'lineitem.tbl.59',
      'lineitem.tbl.60',
      'lineitem.tbl.61',
      'lineitem.tbl.62',
      'lineitem.tbl.63',
      'lineitem.tbl.64',
      'lineitem.tbl.65',
      'lineitem.tbl.66',
      'lineitem.tbl.67',
      'lineitem.tbl.68',
      'lineitem.tbl.69',
      'lineitem.tbl.70',
      'lineitem.tbl.71',
      'lineitem.tbl.72',
      'lineitem.tbl.73',
      'lineitem.tbl.74',
      'lineitem.tbl.75',
      'lineitem.tbl.76',
      'lineitem.tbl.77',
      'lineitem.tbl.78',
      'lineitem.tbl.79',
      'lineitem.tbl.80',
      'lineitem.tbl.81',
      'lineitem.tbl.82',
      'lineitem.tbl.83',
      'lineitem.tbl.84'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table o_et;
create table o_et(
   o_orderkey          number ,
   o_custkey           number ,
   o_orderstatus       char(1) ,
   o_totalprice        number ,
   o_orderdate         date ,
   o_orderpriority     char(15) ,
```

```
   o_clerk          char(15) ,
   o_shippriority   number ,
   o_comment        varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                    records delimited by newline
                    badfile 'o_et.bad'
                    logfile 'o_et.log'
                    fields terminated by '|'
                    missing field values are null
          )
          location (
      'orders.tbl.1',
      'orders.tbl.2',
      'orders.tbl.3',
      'orders.tbl.4',
      'orders.tbl.5',
      'orders.tbl.6',
      'orders.tbl.7',
      'orders.tbl.8',
      'orders.tbl.9',
      'orders.tbl.10',
      'orders.tbl.11',
      'orders.tbl.12',
      'orders.tbl.13',
      'orders.tbl.14',
      'orders.tbl.15',
      'orders.tbl.16',
      'orders.tbl.17',
      'orders.tbl.18',
      'orders.tbl.19',
      'orders.tbl.20',
      'orders.tbl.21',
      'orders.tbl.22',
      'orders.tbl.23',
      'orders.tbl.24',
      'orders.tbl.25',
      'orders.tbl.26',
      'orders.tbl.27',
      'orders.tbl.28',
      'orders.tbl.29',
      'orders.tbl.30',
      'orders.tbl.31',
      'orders.tbl.32',
      'orders.tbl.33',
      'orders.tbl.34',
      'orders.tbl.35',
      'orders.t bl.36',
      'orders.tbl.37',
      'orders.tbl.38',
      'orders.tbl.39',
      'orders.tbl.40',
      'orders.tbl.41',
      'orders.tbl.42',
      'orders.tbl.43',
      'orders.tbl.44',
      'orders.tbl.45',
      'orders.tbl.46',
      'orders.tbl.47',
      'orders.tbl.48',
      'orders.tbl.49',
      'orders.tbl.50',
      'orders.tbl.51',
      'orders.tbl.52',
      'orders.tbl.53',
      'orders.tbl.54',
      'orders.tbl.55',
      'orders.tbl.56',
      'orders.tbl.57',
      'orders.tbl.58',
      'orders.tbl.59',
      'orders.tbl.60',
      'orders.tbl.61',
      'orders.tbl.62',
      'orders.tbl.63',
      'orders.tbl.64',
      'orders.tbl.65',
      'orders.tbl.66',
      'orders.tbl.67',
      'orders.tbl.68',
      'orders.tbl.69',
      'orders.tbl.70',
      'orders.tbl.71',
      'orders.tbl.72',
      'orders.tbl.73',
      'orders.tbl.74',
      'orders.tbl.75',
      'orders.tbl.76',
      'orders.tbl.77',
      'orders.tbl.78',
      'orders.tbl.79',
      'orders.tbl.80',
      'orders.tbl.81',
      'orders.tbl.82',
      'orders.tbl.83',
      'orders.tbl.84'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table ps_et;
create table ps_et(
   ps_partkey       number ,
   ps_suppkey       number ,
   ps_availqty      number ,
   ps_supplycost    number ,
   ps_comment       varchar(199)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                    records delimited by newline
                    badfile 'ps_et.bad'
                    logfile 'ps_et.log'
                    fields terminated by '|'
                    missing field values are null
          )
          location (
      'partsupp.tbl.1',
      'partsupp.tbl.2',
      'partsupp.tbl.3',
      'partsupp.tbl.4',
      'partsupp.tbl.5',
      'partsupp.tbl.6',
      'partsupp.tbl.7',
      'partsupp.tbl.8',
      'partsupp.tbl.9',
      'partsupp.tbl.10',
      'partsupp.tbl.11',
      'partsupp.tbl.12',
      'partsupp.tbl.13',
      'partsupp.tbl.14',
```

```
        'partsupp.tbl.15',
        'partsupp.tbl.16',
        'partsupp.tbl.17',
        'partsupp.tbl.18',
        'partsupp.tbl.19',
        'partsupp.tbl.20',
        'partsupp.tbl.21',
        'partsupp.tbl.22',
        'partsupp.tbl.23',
        'partsupp.tbl.24',
        'partsupp.tbl.25',
        'partsupp.tbl.26',
        'partsupp.tbl.27',
        'partsupp.tbl.28',
        'partsupp.tbl.29',
        'partsupp.tbl.30',
        'partsupp.tbl.31',
        'partsupp.tbl.32',
        'partsupp.tbl.33',
        'partsupp.tbl.34',
        'partsupp.tbl.35',
        'partsupp.tbl.36',
        'partsupp.tbl.37',
        'partsupp.tbl.38',
        'partsupp.tbl.39',
        'partsupp.tbl.40',
        'partsupp.tbl.41',
        'partsupp.tbl.42',
        'partsupp.tbl.43',
        'partsupp.tbl.44',
        'partsupp.tbl.45',
        'partsupp.tbl.46',
        'partsupp.tbl.47',
        'partsupp.tbl.48',
        'partsupp.tbl.49',
        'partsupp.tbl.50',
        'partsupp.tbl.51',
        'partsupp.tbl.52',
        'partsupp.tbl.53',
        'partsupp.tbl.54',
        'partsupp.tbl.55',
        'partsupp.tbl.56',
        'partsupp.tbl.57',
        'partsupp.tbl.58',
        'partsupp.tbl.59',
        'partsupp.tbl.60',
        'partsupp.tbl.61',
        'partsupp.tbl.62',
        'partsupp.tbl.63',
        'partsupp.tbl.64'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table p_et;
create table p_et(
   p_partkey          number ,
   p_name             varchar(55) ,
   p_mfgr             char(25) ,
   p_brand            char(10) ,
   p_type             varchar(25) ,
   p_size             number ,
   p_container        char(10) ,
   p_retailprice      number ,
   p_comment          varchar(23)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        badfile 'p_et.bad'
                        logfile 'p_et.log'
                        fields terminated by '|'
                        missing field values are null
                )
                location (
        'part.tbl.1',
        'part.tbl.2',
        'part.tbl.3',
        'part.tbl.4',
        'part.tbl.5',
        'part.tbl.6',
        'part.tbl.7',
        'part.tbl.8',
        'part.tbl.9',
        'part.tbl.10',
        'part.tbl.11',
        'part.tbl.12',
        'part.tbl.13',
        'part.tbl.14',
        'part.tbl.15',
        'part.tbl.16'
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table c_et;
create table c_et(
   c_custkey          number ,
   c_name             varchar(25) ,
   c_address          varchar(40) ,
   c_nationkey        number ,
   c_phone            char(15) ,
   c_acctbal          number ,
   c_mktsegment       char(10) ,
   c_comment          varchar(117)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        badfile 'c_et.bad'
                        logfile 'c_et.log'
                        fields terminated by '|'
                        missing field values are null
                )
                location (
        'customer.tbl.1',
        'customer.tbl.2',
        'customer.tbl.3',
        'customer.tbl.4',
        'customer.tbl.5',
        'customer.tbl.6',
        'customer.tbl.7',
        'customer.tbl.8',
        'customer.tbl.9',
        'customer.tbl.10',
        'customer.tbl.11',
        'customer.tbl.12',
        'customer.tbl.13',
        'customer.tbl.14',
        'customer.tbl.15',
        'customer.tbl.16'
```

```
))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table s_et;
create table s_et(
   s_suppkey          number ,
   s_name             char(25) ,
   s_address          varchar(40) ,
   s_nationkey        number ,
   s_phone            char(15) ,
   s_acctbal          number ,
   s_comment          varchar(101)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        badfile 's_et.bad'
                        logfile 's_et.log'
                        fields terminated by '|'
                        missing field values are null
            )
            location (
'supplier.tbl'
            ))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table n_et;
create table n_et(
   n_nationkey        number ,
   n_name             char(25) ,
   n_regionkey        number ,
   n_comment          varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        badfile 'n_et.bad'
                        logfile 'n_et.log'
                        fields terminated by '|'
                        missing field values are null
            )
            location (
            'nation.tbl'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
drop table r_et;
create table r_et(
   r_regionkey        number ,
   r_name             char(25) ,
   r_comment          varchar(152)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
                        records delimited by newline
                        badfile 'r_et.bad'
                        logfile 'r_et.log'
                        fields terminated by '|'
                        missing field values are null
            )
            location (
            'region.tbl'))
reject limit unlimited;
}
*sql
{
connect tpch/tpch;
alter table l_et parallel;
alter table o_et parallel;
alter table ps_et parallel;
alter table p_et parallel;
alter table c_et parallel;
alter table s_et parallel;
}
# altering tpch's default and temporary tablespace
*sql
{
alter user tpch default tablespace ts_default;
alter user tpch temporary tablespace ts_temp;
}
*sql
{
connect tpch/tpch
@?/rdbms/admin/utlxplan.sql;
}
*wait
*sql
{
set timing on
set echo on
!date
connect tpch/tpch;
connect tpch/tpch;
rem drop table lineitem;
create table lineitem(
   l_shipdate         ,
   l_orderkey         NOT NULL,
   l_discount         NOT NULL,
   l_extendedprice    NOT NULL,
   l_suppkey          NOT NULL,
   l_quantity         NOT NULL,
   l_returnflag       ,
   l_partkey          NOT NULL,
   l_linestatus       ,
   l_tax              NOT NULL,
   l_commitdate       ,
   l_receiptdate      ,
   l_shipmode         ,
   l_linenumber       NOT NULL,
   l_shipinstruct     ,
   l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 260m next 260m freelist groups 4 freelists 99)
parallel
nologging
partition by range (l_shipdate)
subpartition by hash(l_partkey)
subpartitions 16
(
```

```
partition item1 values less than (to_date('1992-01-01','YYYY-MM-
DD'))
store in (ts_l1,ts_l2,ts_l3,ts_l4)
,
partition item2 values less than (to_date('1992-02-01','YYYY-MM-
DD'))
store in (ts_l5,ts_l6,ts_l7,ts_l8)
,
partition item3 values less than (to_date('1992-03-01','YYYY-MM-
DD'))
store in (ts_l9,ts_l10,ts_l11,ts_l12)
,
partition item4 values less than (to_date('1992-04-01','YYYY-MM-
DD'))
store in (ts_l13,ts_l14,ts_l15,ts_l16)
,
partition item5 values less than (to_date('1992-05-01','YYYY-MM-
DD'))
store in (ts_l17,ts_l18,ts_l19,ts_l20)
,
partition item6 values less than (to_date('1992-06-01','YYYY-MM-
DD'))
store in (ts_l21,ts_l22,ts_l23,ts_l24)
,
partition item7 values less than (to_date('1992-07-01','YYYY-MM-
DD'))
store in (ts_l25,ts_l26,ts_l27,ts_l28)
,
partition item8 values less than (to_date('1992-08-01','YYYY-MM-
DD'))
store in (ts_l29,ts_l30,ts_l31,ts_l32)
,
partition item9 values less than (to_date('1992-09-01','YYYY-MM-
DD'))
store in (ts_l33,ts_l34,ts_l35,ts_l36)
,
partition item10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store in (ts_l37,ts_l38,ts_l39,ts_l40)
,
partition item11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store in (ts_l41,ts_l42,ts_l43,ts_l44)
,
partition item12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store in (ts_l45,ts_l46,ts_l47,ts_l48)
,
partition item13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store in (ts_l49,ts_l50,ts_l51,ts_l52)
,
partition item14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store in (ts_l53,ts_l54,ts_l55,ts_l56)
,
partition item15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store in (ts_l57,ts_l58,ts_l59,ts_l60)
,
partition item16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store in (ts_l61,ts_l62,ts_l63,ts_l64)
,
partition item17 values less than (to_date('1993-05-01','YYYY-MM-
DD'))
store in (ts_l65,ts_l66,ts_l67,ts_l68)
,
partition item18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store in (ts_l69,ts_l70,ts_l71,ts_l72)
,
partition item19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store in (ts_l73,ts_l74,ts_l75,ts_l76)
,
partition item20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
store in (ts_l77,ts_l78,ts_l79,ts_l80)
,
partition item21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store in (ts_l81,ts_l82,ts_l83,ts_l84)
,
partition item22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store in (ts_l85,ts_l86,ts_l87,ts_l88)
,
partition item23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store in (ts_l89,ts_l90,ts_l91,ts_l92)
,
partition item24 values less than (to_date('1993-12-01','YYYY-MM-
DD'))
store in (ts_l93,ts_l94,ts_l95,ts_l96)
,
partition item25 values less than (to_date('1994-01-01','YYYY-MM-
DD'))
store in (ts_l97,ts_l98,ts_l99,ts_l100)
,
partition item26 values less than (to_date('1994-02-01','YYYY-MM-
DD'))
store in (ts_l101,ts_l102,ts_l103,ts_l104)
,
partition item27 values less than (to_date('1994-03-01','YYYY-MM-
DD'))
store in (ts_l105,ts_l106,ts_l107,ts_l108)
,
partition item28 values less than (to_date('1994-04-01','YYYY-MM-
DD'))
store in (ts_l109,ts_l110,ts_l111,ts_l112)
,
partition item29 values less than (to_date('1994-05-01','YYYY-MM-
DD'))
store in (ts_l113,ts_l114,ts_l115,ts_l116)
,
partition item30 values less than (to_date('1994-06-01','YYYY-MM-
DD'))
store in (ts_l117,ts_l118,ts_l119,ts_l120)
,
partition item31 values less than (to_date('1994-07-01','YYYY-MM-
DD'))
store in (ts_l121,ts_l122,ts_l123,ts_l124)
,
partition item32 values less than (to_date('1994-08-01','YYYY-MM-
DD'))
store in (ts_l125,ts_l126,ts_l127,ts_l128)
,
partition item33 values less than (to_date('1994-09-01','YYYY-MM-
DD'))
store in (ts_l129,ts_l130,ts_l131,ts_l132)
,
partition item34 values less than (to_date('1994-10-01','YYYY-MM-
DD'))
store in (ts_l133,ts_l134,ts_l135,ts_l136)
,
partition item35 values less than (to_date('1994-11-01','YYYY-MM-
DD'))
store in (ts_l137,ts_l138,ts_l139,ts_l140)
,
```

```
partition item36 values less than (to_date('1994-12-01','YYYY-MM-
DD'))
store in (ts_l141,ts_l142,ts_l143,ts_l144)
,
partition item37 values less than (to_date('1995-01-01','YYYY-MM-
DD'))
store in (ts_l145,ts_l146,ts_l147,ts_l148)
,
partition item38 values less than (to_date('1995-02-01','YYYY-MM-
DD'))
store in (ts_l149,ts_l150,ts_l151,ts_l152)
,
partition item39 values less than (to_date('1995-03-01','YYYY-MM-
DD'))
store in (ts_l153,ts_l154,ts_l155,ts_l156)
,
partition item40 values less than (to_date('1995-04-01','YYYY-MM-
DD'))
store in (ts_l157,ts_l158,ts_l159,ts_l160)
,
partition item41 values less than (to_date('1995-05-01','YYYY-MM-
DD'))
store in (ts_l161,ts_l162,ts_l163,ts_l164)
,
partition item42 values less than (to_date('1995-06-01','YYYY-MM-
DD'))
store in (ts_l165,ts_l166,ts_l167,ts_l168)
,
partition item43 values less than (to_date('1995-07-01','YYYY-MM-
DD'))
store in (ts_l169,ts_l170,ts_l171,ts_l172)
,
partition item44 values less than (to_date('1995-08-01','YYYY-MM-
DD'))
store in (ts_l173,ts_l174,ts_l175,ts_l176)
,
partition item45 values less than (to_date('1995-09-01','YYYY-MM-
DD'))
store in (ts_l177,ts_l178,ts_l179,ts_l180)
,
partition item46 values less than (to_date('1995-10-01','YYYY-MM-
DD'))
store in (ts_l181,ts_l182,ts_l183,ts_l184)
,
partition item47 values less than (to_date('1995-11-01','YYYY-MM-
DD'))
store in (ts_l185,ts_l186,ts_l187,ts_l188)
,
partition item48 values less than (to_date('1995-12-01','YYYY-MM-
DD'))
store in (ts_l189,ts_l190,ts_l191,ts_l192)
,
partition item49 values less than (to_date('1996-01-01','YYYY-MM-
DD'))
store in (ts_l193,ts_l194,ts_l195,ts_l196)
,
partition item50 values less than (to_date('1996-02-01','YYYY-MM-
DD'))
store in (ts_l197,ts_l198,ts_l199,ts_l200)
,
partition item51 values less than (to_date('1996-03-01','YYYY-MM-
DD'))
store in (ts_l201,ts_l202,ts_l203,ts_l204)
,
partition item52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store in (ts_l205,ts_l206,ts_l207,ts_l208)
,
partition item53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store in (ts_l209,ts_l210,ts_l211,ts_l212)
,
partition item54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store in (ts_l213,ts_l214,ts_l215,ts_l216)
,
partition item55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store in (ts_l217,ts_l218,ts_l219,ts_l220)
,
partition item56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store in (ts_l221,ts_l222,ts_l223,ts_l224)
,
partition item57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store in (ts_l225,ts_l226,ts_l227,ts_l228)
,
partition item58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store in (ts_l229,ts_l230,ts_l231,ts_l232)
,
partition item59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store in (ts_l233,ts_l234,ts_l235,ts_l236)
,
partition item60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store in (ts_l237,ts_l238,ts_l239,ts_l240)
,
partition item61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store in (ts_l241,ts_l242,ts_l243,ts_l244)
,
partition item62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store in (ts_l245,ts_l246,ts_l247,ts_l248)
,
partition item63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
store in (ts_l249,ts_l250,ts_l251,ts_l252)
,
partition item64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store in (ts_l253,ts_l254,ts_l255,ts_l256)
,
partition item65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store in (ts_l257,ts_l258,ts_l259,ts_l260)
,
partition item66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store in (ts_l261,ts_l262,ts_l263,ts_l264)
,
partition item67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store in (ts_l265,ts_l266,ts_l267,ts_l268)
,
partition item68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store in (ts_l269,ts_l270,ts_l271,ts_l272)
,
partition item69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store in (ts_l273,ts_l274,ts_l275,ts_l276)
,
partition item70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store in (ts_l277,ts_l278,ts_l279,ts_l280)
,
```

```
partition item71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store in (ts_l281,ts_l282,ts_l283,ts_l284)
,
partition item72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store in (ts_l285,ts_l286,ts_l287,ts_l288)
,
partition item73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store in (ts_l289,ts_l290,ts_l291,ts_l292)
,
partition item74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store in (ts_l293,ts_l294,ts_l295,ts_l296)
,
partition item75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store in (ts_l297,ts_l298,ts_l299,ts_l300)
,
partition item76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store in (ts_l301,ts_l302,ts_l303,ts_l304)
,
partition item77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store in (ts_l305,ts_l306,ts_l307,ts_l308)
,
partition item78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store in (ts_l309,ts_l310,ts_l311,ts_l312)
,
partition item79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store in (ts_l313,ts_l314,ts_l315,ts_l316)
,
partition item80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store in (ts_l317,ts_l318,ts_l319,ts_l320)
,
partition item81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store in (ts_l321,ts_l322,ts_l323,ts_l324)
,
partition item82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store in (ts_l325,ts_l326,ts_l327,ts_l328)
,
partition item83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
store in (ts_l329,ts_l330,ts_l331,ts_l332)
,
partition item84 values less than (MAXVALUE)
store in (ts_l333,ts_l334,ts_l335,ts_l336) )
as select
   l_shipdate      ,
   l_orderkey      ,
   l_discount      ,
   l_extendedprice ,
   l_suppkey       ,
   l_quantity      ,
   l_returnflag    ,
   l_partkey       ,
   l_linestatus    ,
   l_tax           ,
   l_commitdate    ,
   l_receiptdate   ,
   l_shipmode      ,
   l_linenumber    ,
   l_shipinstruct  ,
   l_comment
from l_et;
!date
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
!date

rem drop table orders;
create table orders(
   o_orderdate     ,
   o_orderkey      NOT NULL,
   o_custkey       NOT NULL,
   o_orderpriority ,
   o_shippriority  ,
   o_clerk         ,
   o_orderstatus   ,
   o_totalprice    ,
   o_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 115m next 115m freelist groups 4 freelists 99)
parallel
nologging
partition by range (o_orderdate)
subpartition by hash(o_custkey)
subpartitions 16
(
partition ord1 values less than (to_date('1992-01-01','YYYY-MM-
DD'))
store in (ts_o1,ts_o2,ts_o3,ts_o4)
,
partition ord2 values less than (to_date('1992-02-01','YYYY-MM-
DD'))
store in (ts_o5,ts_o6,ts_o7,ts_o8)
,
partition ord3 values less than (to_date('1992-03-01','YYYY-MM-
DD'))
store in (ts_o9,ts_o10,ts_o11,ts_o12)
,
partition ord4 values less than (to_date('1992-04-01','YYYY-MM-
DD'))
store in (ts_o13,ts_o14,ts_o15,ts_o16)
,
partition ord5 values less than (to_date('1992-05-01','YYYY-MM-
DD'))
store in (ts_o17,ts_o18,ts_o19,ts_o20)
,
partition ord6 values less than (to_date('1992-06-01','YYYY-MM-
DD'))
store in (ts_o21,ts_o22,ts_o23,ts_o24)
,
partition ord7 values less than (to_date('1992-07-01','YYYY-MM-
DD'))
store in (ts_o25,ts_o26,ts_o27,ts_o28)
,
partition ord8 values less than (to_date('1992-08-01','YYYY-MM-
DD'))
store in (ts_o29,ts_o30,ts_o31,ts_o32)
,
partition ord9 values less than (to_date('1992-09-01','YYYY-MM-
DD'))
store in (ts_o33,ts_o34,ts_o35,ts_o36)
,
```

```
partition ord10 values less than (to_date('1992-10-01','YYYY-MM-
DD'))
store in (ts_o37,ts_o38,ts_o39,ts_o40)
,
partition ord11 values less than (to_date('1992-11-01','YYYY-MM-
DD'))
store in (ts_o41,ts_o42,ts_o43,ts_o44)
,
partition ord12 values less than (to_date('1992-12-01','YYYY-MM-
DD'))
store in (ts_o45,ts_o46,ts_o47,ts_o48)
,
partition ord13 values less than (to_date('1993-01-01','YYYY-MM-
DD'))
store in (ts_o49,ts_o50,ts_o51,ts_o52)
,
partition ord14 values less than (to_date('1993-02-01','YYYY-MM-
DD'))
store in (ts_o53,ts_o54,ts_o55,ts_o56)
,
partition ord15 values less than (to_date('1993-03-01','YYYY-MM-
DD'))
store in (ts_o57,ts_o58,ts_o59,ts_o60)
,
partition ord16 values less than (to_date('1993-04-01','YYYY-MM-
DD'))
store in (ts_o61,ts_o62,ts_o63,ts_o64)
,
partition ord17 values less than (to_date('1993-05-01','YYYY-MM-
DD'))
store in (ts_o65,ts_o66,ts_o67,ts_o68)
,
partition ord18 values less than (to_date('1993-06-01','YYYY-MM-
DD'))
store in (ts_o69,ts_o70,ts_o71,ts_o72)
,
partition ord19 values less than (to_date('1993-07-01','YYYY-MM-
DD'))
store in (ts_o73,ts_o74,ts_o75,ts_o76)
,
partition ord20 values less than (to_date('1993-08-01','YYYY-MM-
DD'))
store in (ts_o77,ts_o78,ts_o79,ts_o80)
,
partition ord21 values less than (to_date('1993-09-01','YYYY-MM-
DD'))
store in (ts_o81,ts_o82,ts_o83,ts_o84)
,
partition ord22 values less than (to_date('1993-10-01','YYYY-MM-
DD'))
store in (ts_o85,ts_o86,ts_o87,ts_o88)
,
partition ord23 values less than (to_date('1993-11-01','YYYY-MM-
DD'))
store in (ts_o89,ts_o90,ts_o91,ts_o92)
,
partition ord24 values less than (to_date('1993-12-01','YYYY-MM-
DD'))
store in (ts_o93,ts_o94,ts_o95,ts_o96)
,
partition ord25 values less than (to_date('1994-01-01','YYYY-MM-
DD'))
store in (ts_o97,ts_o98,ts_o99,ts_o100)
,
partition ord26 values less than (to_date('1994-02-01','YYYY-MM-
DD'))
store in (ts_o101,ts_o102,ts_o103,ts_o104)
,
partition ord27 values less than (to_date('1994-03-01','YYYY-MM-
DD'))
store in (ts_o105,ts_o106,ts_o107,ts_o108)
,
partition ord28 values less than (to_date('1994-04-01','YYYY-MM-
DD'))
store in (ts_o109,ts_o110,ts_o111,ts_o112)
,
partition ord29 values less than (to_date('1994-05-01','YYYY-MM-
DD'))
store in (ts_o113,ts_o114,ts_o115,ts_o116)
,
partition ord30 values less than (to_date('1994-06-01','YYYY-MM-
DD'))
store in (ts_o117,ts_o118,ts_o119,ts_o120)
,
partition ord31 values less than (to_date('1994-07-01','YYYY-MM-
DD'))
store in (ts_o121,ts_o122,ts_o123,ts_o124)
,
partition ord32 values less than (to_date('1994-08-01','YYYY-MM-
DD'))
store in (ts_o125,ts_o126,ts_o127,ts_o128)
,
partition ord33 values less than (to_date('1994-09-01','YYYY-MM-
DD'))
store in (ts_o129,ts_o130,ts_o131,ts_o132)
,
partition ord34 values less than (to_date('1994-10-01','YYYY-MM-
DD'))
store in (ts_o133,ts_o134,ts_o135,ts_o136)
,
partition ord35 values less than (to_date('1994-11-01','YYYY-MM-
DD'))
store in (ts_o137,ts_o138,ts_o139,ts_o140)
,
partition ord36 values less than (to_date('1994-12-01','YYYY-MM-
DD'))
store in (ts_o141,ts_o142,ts_o143,ts_o144)
,
partition ord37 values less than (to_date('1995-01-01','YYYY-MM-
DD'))
store in (ts_o145,ts_o146,ts_o147,ts_o148)
,
partition ord38 values less than (to_date('1995-02-01','YYYY-MM-
DD'))
store in (ts_o149,ts_o150,ts_o151,ts_o152)
,
partition ord39 values less than (to_date('1995-03-01','YYYY-MM-
DD'))
store in (ts_o153,ts_o154,ts_o155,ts_o156)
,
partition ord40 values less than (to_date('1995-04-01','YYYY-MM-
DD'))
store in (ts_o157,ts_o158,ts_o159,ts_o160)
,
partition ord41 values less than (to_date('1995-05-01','YYYY-MM-
DD'))
store in (ts_o161,ts_o162,ts_o163,ts_o164)
,
partition ord42 values less than (to_date('1995-06-01','YYYY-MM-
DD'))
store in (ts_o165,ts_o166,ts_o167,ts_o168)
,
partition ord43 values less than (to_date('1995-07-01','YYYY-MM-
DD'))
store in (ts_o169,ts_o170,ts_o171,ts_o172)
,
partition ord44 values less than (to_date('1995-08-01','YYYY-MM-
DD'))
store in (ts_o173,ts_o174,ts_o175,ts_o176)
,
```

```
partition ord45 values less than (to_date('1995-09-01','YYYY-MM-
DD'))
store in (ts_o177,ts_o178,ts_o179,ts_o180)
,
partition ord46 values less than (to_date('1995-10-01','YYYY-MM-
DD'))
store in (ts_o181,ts_o182,ts_o183,ts_o184)
,
partition ord47 values less than (to_date('1995-11-01','YYYY-MM-
DD'))
store in (ts_o185,ts_o186,ts_o187,ts_o188)
,
partition ord48 values less than (to_date('1995-12-01','YYYY-MM-
DD'))
store in (ts_o189,ts_o190,ts_o191,ts_o192)
,
partition ord49 values less than (to_date('1996-01-01','YYYY-MM-
DD'))
store in (ts_o193,ts_o194,ts_o195,ts_o196)
,
partition ord50 values less than (to_date('1996-02-01','YYYY-MM-
DD'))
store in (ts_o197,ts_o198,ts_o199,ts_o200)
,
partition ord51 values less than (to_date('1996-03-01','YYYY-MM-
DD'))
store in (ts_o201,ts_o202,ts_o203,ts_o204)
,
partition ord52 values less than (to_date('1996-04-01','YYYY-MM-
DD'))
store in (ts_o205,ts_o206,ts_o207,ts_o208)
,
partition ord53 values less than (to_date('1996-05-01','YYYY-MM-
DD'))
store in (ts_o209,ts_o210,ts_o211,ts_o212)
,
partition ord54 values less than (to_date('1996-06-01','YYYY-MM-
DD'))
store in (ts_o213,ts_o214,ts_o215,ts_o216)
,
partition ord55 values less than (to_date('1996-07-01','YYYY-MM-
DD'))
store in (ts_o217,ts_o218,ts_o219,ts_o220)
,
partition ord56 values less than (to_date('1996-08-01','YYYY-MM-
DD'))
store in (ts_o221,ts_o222,ts_o223,ts_o224)
,
partition ord57 values less than (to_date('1996-09-01','YYYY-MM-
DD'))
store in (ts_o225,ts_o226,ts_o227,ts_o228)
,
partition ord58 values less than (to_date('1996-10-01','YYYY-MM-
DD'))
store in (ts_o229,ts_o230,ts_o231,ts_o232)
,
partition ord59 values less than (to_date('1996-11-01','YYYY-MM-
DD'))
store in (ts_o233,ts_o234,ts_o235,ts_o236)
,
partition ord60 values less than (to_date('1996-12-01','YYYY-MM-
DD'))
store in (ts_o237,ts_o238,ts_o239,ts_o240)
,
partition ord61 values less than (to_date('1997-01-01','YYYY-MM-
DD'))
store in (ts_o241,ts_o242,ts_o243,ts_o244)
,
partition ord62 values less than (to_date('1997-02-01','YYYY-MM-
DD'))
store in (ts_o245,ts_o246,ts_o247,ts_o248)
,
partition ord63 values less than (to_date('1997-03-01','YYYY-MM-
DD'))
store in (ts_o249,ts_o250,ts_o251,ts_o252)
,
partition ord64 values less than (to_date('1997-04-01','YYYY-MM-
DD'))
store in (ts_o253,ts_o254,ts_o255,ts_o256)
,
partition ord65 values less than (to_date('1997-05-01','YYYY-MM-
DD'))
store in (ts_o257,ts_o258,ts_o259,ts_o260)
,
partition ord66 values less than (to_date('1997-06-01','YYYY-MM-
DD'))
store in (ts_o261,ts_o262,ts_o263,ts_o264)
,
partition ord67 values less than (to_date('1997-07-01','YYYY-MM-
DD'))
store in (ts_o265,ts_o266,ts_o267,ts_o268)
,
partition ord68 values less than (to_date('1997-08-01','YYYY-MM-
DD'))
store in (ts_o269,ts_o270,ts_o271,ts_o272)
,
partition ord69 values less than (to_date('1997-09-01','YYYY-MM-
DD'))
store in (ts_o273,ts_o274,ts_o275,ts_o276)
,
partition ord70 values less than (to_date('1997-10-01','YYYY-MM-
DD'))
store in (ts_o277,ts_o278,ts_o279,ts_o280)
,
partition ord71 values less than (to_date('1997-11-01','YYYY-MM-
DD'))
store in (ts_o281,ts_o282,ts_o283,ts_o284)
,
partition ord72 values less than (to_date('1997-12-01','YYYY-MM-
DD'))
store in (ts_o285,ts_o286,ts_o287,ts_o288)
,
partition ord73 values less than (to_date('1998-01-01','YYYY-MM-
DD'))
store in (ts_o289,ts_o290,ts_o291,ts_o292)
,
partition ord74 values less than (to_date('1998-02-01','YYYY-MM-
DD'))
store in (ts_o293,ts_o294,ts_o295,ts_o296)
,
partition ord75 values less than (to_date('1998-03-01','YYYY-MM-
DD'))
store in (ts_o297,ts_o298,ts_o299,ts_o300)
,
partition ord76 values less than (to_date('1998-04-01','YYYY-MM-
DD'))
store in (ts_o301,ts_o302,ts_o303,ts_o304)
,
partition ord77 values less than (to_date('1998-05-01','YYYY-MM-
DD'))
store in (ts_o305,ts_o306,ts_o307,ts_o308)
,
partition ord78 values less than (to_date('1998-06-01','YYYY-MM-
DD'))
store in (ts_o309,ts_o310,ts_o311,ts_o312)
,
partition ord79 values less than (to_date('1998-07-01','YYYY-MM-
DD'))
store in (ts_o313,ts_o314,ts_o315,ts_o316)
,
```

```
partition ord80 values less than (to_date('1998-08-01','YYYY-MM-
DD'))
store in (ts_o317,ts_o318,ts_o319,ts_o320)
,
partition ord81 values less than (to_date('1998-09-01','YYYY-MM-
DD'))
store in (ts_o321,ts_o322,ts_o323,ts_o324)
,
partition ord82 values less than (to_date('1998-10-01','YYYY-MM-
DD'))
store in (ts_o325,ts_o326,ts_o327,ts_o328)
,
partition ord83 values less than (to_date('1998-11-01','YYYY-MM-
DD'))
store in (ts_o329,ts_o330,ts_o331,ts_o332)
,
partition ord84 values less than (MAXVALUE)
store in (ts_o333,ts_o334,ts_o335,ts_o336) )
as select
   o_orderdate        ,
   o_orderkey         ,
   o_custkey          ,
   o_orderpriority    ,
   o_shippriority     ,
   o_clerk            ,
   o_orderstatus      ,
   o_totalprice       ,
   o_comment
from o_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date

rem drop table partsupp;
create table partsupp(
   ps_partkey          NOT NULL,
   ps_suppkey          NOT NULL,
   ps_supplycost       NOT NULL,
   ps_availqty        ,
   ps_comment         ,
constraint pk_partkey_suppkey_1 primary key(ps_partkey,
ps_suppkey)
)
organization index
partition by hash(ps_partkey)
partitions 128
storage (initial 335m next 335m)
parallel
nologging
pctthreshold 50
tablespace ts_psupp1
as select
   ps_partkey         ,
   ps_suppkey         ,
   ps_supplycost      ,
   ps_availqty        ,
   ps_comment
from ps_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
rem drop table customer;
create table customer(
   c_custkey          NOT NULL,
   c_mktsegment       ,
   c_nationkey        ,
   c_name             ,
   c_address          ,
   c_phone            ,
   c_acctbal          ,
   c_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 75m next 75m)
partition by hash (c_custkey)
partitions 128
store in
(ts_small1,ts_small2,ts_small3,ts_small4,ts_small5,ts_small6,ts_sma
ll7,ts_small8,ts_small9,ts_small10,ts_small11,ts_small12,ts_small13
,ts_small14,ts_small15,ts_small16)
as select
   c_custkey          ,
   c_mktsegment       ,
   c_nationkey        ,
   c_name             ,
   c_address          ,
   c_phone            ,
   c_acctbal          ,
   c_comment
from c_et;
!date
}
*wait
*sql
{
connect tpch/tpch
set timing on
set echo on

!date
rem drop table part;

create table part(
   p_partkey          NOT NULL,
   p_type             ,
   p_size             ,
   p_brand            ,
   p_name             ,
   p_container        ,
   p_mfgr             ,
   p_retailprice      ,
   p_comment
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 85m next 85m)
partition by hash (p_partkey)
partitions 128
store in
(ts_small1,ts_small2,ts_small3,ts_small4,ts_small5,ts_small6,ts_sma
```

```
ll7,ts_small8,ts_small9,ts_small10,ts_small11,ts_small12,ts_small13
,ts_small14,ts_small15,ts_small16)
as select
   p_partkey          ,
   p_type             ,
   p_size             ,
   p_brand            ,
   p_name             ,
   p_container        ,
   p_mfgr             ,
   p_retailprice      ,
   p_comment
from p_et;
!date
}
*wait
*sql
{
connect tpch/tpch;
set timing on
set echo on
rem drop table supplier;
create table supplier(
   s_suppkey          NOT NULL,
   s_nationkey        ,
   s_comment          ,
   s_name             ,
   s_address          ,
   s_phone            ,
   s_acctbal
)
pctfree 0
pctused 99
parallel
nologging
storage (initial 40m next 5m)
partition by hash (s_suppkey)
partitions 128
store in
(ts_small1,ts_small2,ts_small3,ts_small4,ts_small5,ts_small6,ts_sma
ll7,ts_small8,ts_small9,ts_small10,ts_small11,ts_small12,ts_small13
,ts_small14,ts_small15,ts_small16)
as select
   s_suppkey          ,
   s_nationkey        ,
   s_comment          ,
   s_name             ,
   s_address          ,
   s_phone            ,
   s_acctbal
from s_et;
}
*wait
*sql
{
connect tpch/tpch;
set echo on
set timing on

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

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

*wait
*bgoff
%e-scuto

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

!date
drop table l_et;
drop table o_et;
drop table ps_et;
drop table p_et;
drop table c_et;
drop table s_et;
drop table n_et;
drop table r_et;
}
*bgoff
%e-dapop
%b-ixcre
*bgon=1
#######################################################
#####################
# Index Creation Phase
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on
rem drop index i_l_orderkey;
create index i_l_orderkey
on lineitem (l_orderkey)
global partition by hash (l_orderkey)
partitions 128
pctfree 5
initrans 10
tablespace ts_lokey
storage (freelist groups 4 freelists 99)
parallel
compute statistics
nologging;
}
*sql
{
connect tpch/tpch;
!date
set echo on
set timing on
rem drop index i_o_orderkey;
create unique index i_o_orderkey
on orders (o_orderkey)
global partition by hash (o_orderkey)
partitions 128
pctfree 2
initrans 10
tablespace ts_okey
storage (freelist groups 4 freelists 99 )
parallel
compute statistics
```

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

rem drop index i_c_custkey;
create unique index i_c_custkey
on customer (c_custkey)
pctfree 2
initrans 10
tablespace ts_custkey
storage (freelists 99)
parallel
compute statistics
nologging;
}
*wait
*bgoff
%e-ixcre
%b-anlyz
*bgon=1
############################################################
#####################
# Analyze Phase
*sql
{
connect tpch/tpch;
!date
set timing on
execute dbms_stats.gather_schema_stats('TPCH' , estimate_percent
=> 1, degree => 128 , granularity => 'GLOBAL' );
connect / as sysdba
execute dbms_stats.gather_system_stats;
alter system switch logfile;
!date
}
*wait
*bgoff
%e-anlyz
```

B.2 bumpx.pl

```
#!/usr/contrib/bin/perl
#
# $Header: bumpx.pl 13-feb-2001.14:35:03 mpoess Exp $
#
# bumpx.pl
#
#  Copyright (c) Oracle Corporation 1999, 2000. All Rights
Reserved.
#
#    NAME
#      bumpx.pl - <one-line expansion of the name>
#
#    DESCRIPTION
#      <short description of component this file declares/defines>
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      02/13/01 - make global range index possible for
o_orderkey
#    mpoess      01/31/00 - fix bug with undo
#    mpoess      01/21/00 - fix bugs for partitioned tablespaces
#    mpoess      01/20/00 - add OPS load one partiton from one node
support
#    mpoess      01/07/00 - add support for flatfile distribution
#    mpoess      01/06/00 - add controlfile generation performace
improvement
#    mpoess      12/16/99 - add partitioning support for control files
#    mpoess      10/28/99 - adjust bumpx to fixed dbgen (84
partitioning)
#    mpoess      08/17/99 - change column definition of base tables
#    mpoess      08/13/99 - add support for reporting files
#    mpoess      08/08/99 - change file atributes
#    mpoess      07/12/99 - add setpath for dbgen
#    mpoess      07/12/99 - disable -o option for dbgen
#    mpoess      07/07/99 - fix dbgen phase
#    mpoess      07/06/99 - change path to perl
#    mpoess      07/02/99 - add environment variable support for conf
file
#    mpoess      06/28/99 - add pathname for param.txt file
#    mpoess      06/28/99 - Copy from TPCD
#    mpoess      06/28/99 - Creation
#    MODIFIED   (MM/DD/YY)
#    mpoess      11/18/98 - add undo rollback segment support for
OPS systems
#    mpoess      11/18/98 - add control log etc. keywords in SQLLDR
call
#    mpoess      11/06/98 - fix bug in print_as_select
#    mpoess      11/05/98 - change name of BUMPC_CTR variable
#    mpoess      11/04/98 - delete default startup before in dex creation
#    mpoess      11/04/98 - adopt Barry Perkins changes to dbgen
#    mpoess      11/03/98 - change as select handling
#    mpoess      11/02/98 - change background process handling for
NT
#    mpoess      10/26/98 - modifiy dapop so that mult iple partitions
can be lo
#    mpoess      10/23/98 - take *waits out from flat file generation
#    mpoess      10/20/98 - add sdgen, dbgen and dapop description
#    mpoess      10/12/98 - fix bug in constraint creation
#    mpoess      10/12/98 - add support for non partitioned tables
#    mpoess      10/08/98 - add -including new values- clause in view
log defin
#    mpoess      09/27/98 - update links,param2html
#    mpoess      09/25/98 - adding new features
#    akarasik    07/29/98 -
#    akarasik    07/29/98 - Checking in
#     kwong 05/22/98 -  add change storage for tables and indexes in
#                       chdop
#    kwong      05/21/98 -  fix load pipe
#    kwong      05/21/98 -  add some more default values in init.ora
#    kwong      05/20/98 -  add parameter nt_port
#    kwong 05/19/98 -  fix control file generation
#    kwong      05/18/98 -  run utlxplan.sql after created user
#    kwong      05/18/98 -  add "dbgen" phase
#    kwong      05/15/98 -  calculate partition values for l_orderkey
#                       and l_partkey
#    kwong      05/13/98 -  add phase "chdop" for modifying dop as
the
#                       end of the build
#    kwong      05/13/98 -  fix the order of create tablespaces and add
#                       datafiles
#    pswong     04/15/97 -  fix pipeline load
#    pswong     04/02/97 -  analyze partition support
#    pswong     03/27/97 -  introduced ts groups
#    pswong     03/18/97 -  named pipe loader and dbgen support
#    pswong     02/28/97 -  more partition support
#    pswong     02/14/97 -  Version 8
#    amozes     10/27/94 -  Creation

#************************************************************
************************
```

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

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

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

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

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

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

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

&doexecute if $doexecute;

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


exit(0);

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

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

sub docreate
{
   print "Creation pass begun." if $verbose;
   @phases_tmp = @phases; # because I will eat the elements
   while ($phase=shift(@phases_tmp))
   {
      if ($allphases =~ /$phase/)
      {
                  $nextphase = shift(@phases_tmp);
                  unshift (@phases_tmp,$nextphase) if ($nextphase);
                  print "\n  Creating phase \'$phase\'" if $verbose;
         &dump0("%b-$phase\n");
         &prepmulti();
         &dump0("*bgon=$params{$phase.'_max_bg'}");
         eval (&$phase);
```

```
            &dump0("*wait");
            &dump0("*bgoff");
            &dump0("%e-$phase\n");
        }
        else
        {
            print "\n  Phase \'$phase\' not built in...assuming a \*match
block being used";
        }
    }
    print "\nCreation pass complete.\n" if $verbose;
}

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

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

sub execute
```

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

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

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

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

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

            if (($os cmp "unix") == 0)
            {
               if ($bg == 1)
               {
                        print "." if $verbose;
                        $fpid = fork;
                        if ($fpid == 0)
                        {
                           exec ($cmd);
                           print "exec\'d command
failed:\n$cmd\nreason: $!\n";
                           exit(-1);
                        }
                        unshift (@wpids, $fpid);
                        $bgrun = $bgrun + 1;
                        &execute ("*wait") if (($bgrun >= $bgmax)
&& ($bgmax >= 0));
               }
               else
               {
                        system ($cmd);
                        print "system\'d command
failed:\n$cmd\nreason: $? ($!)\n" if $?;
               }
            }
            else ## NT support
            {
```

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

sub usage
{
   print "Usage:\n";
   print " $0 [-c] [-x] [-i infile] [-o outfile] [-p phaselist] [-t type]\n";
   print "   -c : create intermediary file (needed for execution)\n";
   print "   -x : execute intermediary file\n";
   print "   -i : configuration file to be used\n";
   print "        defaults to bumpx.conf in \$BUMPX_DIR or
\$CWD\n";
   print "   -o : intermediary file to be created and/or used\n";
   print "        defaults to bumpx.dat in \$BUMPX_DIR or
\$CWD\n";
   print "   -p : list of phases to create/execute\n";
   print "        phaselist is a comma separated list of phases in
order\n";
   print "        possible phases are:\n";
   print "          sdgen  = seed file generation\n";
   print "          dbgen = data flat file generation\n";
   print "          plcre = NT raw partition and links creation\n";
   print "          dbcre = database creation\n";
   print "          shutd = shutdown database (on all instances) \n";
   print "          start = startup database (on all instances) \n";
   print "          sccre = schema creation \n";
   print "          sctso = schema creation (tablespaces only) \n";
   print "          scuto = schema creation (user and tables only) \n";
   print "          scuvo = schema creation (views only) \n";
   print "          dapop = data population \n";
   print "          ixcre = index creation (including constraints) \n";
   print "          anlyz = analyze objects\n";
   print "          chob  = change parameters of objects\n";
   print "          expln = create explain plans\n";
   print "          query = run and time queries\n";
   print "        defaults to $defphases\n";
   print "   -t : type of benchmark\n";
   print "        enables benchmark-specific defaults\n";
   print "        current possibilties are: $allbmtypes\n";
   print "        defaults to tpcd\n";
   print "   -s : run silent (no parameter checking is done)\n";
   print "\n";
   print "Examples:\n";
   print " $0 -c -o file.out \n";
   print "   This will create an intermediary file named file.out for the
default\n";
   print "   phases using bumpx.conf which can be executed in the
future.\n";
   print " $0 -x -p dapop \n";
   print "   Executes data population phase of intermediary file
bumpx.dat.\n";
   print " $0 -cx -p dbcre,dapop \n";
   print "   This will create an intermediary file, bumpx.dat, using
configuration\n";
   print "   file, bumpx.conf, for both the database creation phase and
the data\n";
   print "   population phase, and then execute that file.\n";
   print "\n";
   print "$error\n";
   exit(-1);
}

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

sub dbcre
{

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

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

# Create database (system tablespace/dat afile, redo logs, rollback
segments)
   @sys_files = split (/,/, $params{'ts_sys_datafiles'});
```

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

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


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

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

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

sub shutd
{

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

sub start
{
```

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

sub sdgen
{

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

sub dbgen
{

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

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

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

   foreach $table (@load_tablelist)
   {
            @load_flatfile_area_list = split(/,/,
$params{'load_flatfile_area_'.$table}) if (defined
$params{'load_flatfile_area_'.$table});
      $curts = $params{'tab_'.$table.'_ts'};
      $usels = 0;
      $lfexist = 0;
      $bfexist = 0;
      $dafexist = 0;
      $difexist = 0;
      $ffexist = 0;

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

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

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

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

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

            $nperset = 1;

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

```
            if (defined @load_flatfile_area_list) {
              $number_of_flatfile_areas = @load_flatfile_area_list;
              if ($number_of_flatfile_areas < $dbgpar) {
                 print "load_flatfile_area_list has to few entries\n";
                 print "set it to at  least ". $dbgpar." elements\n";
                 print "baling out ...\n";
        }
            }
            for ($i=0;$i<$dbgpar;$i++)
            {
               # create script to generate flat files for each dbgpar
               $cmd = sprintf("%s -S %d", $dbgopt, ($i + 1));
               if (defined $params{'load_flatfile_area_'.$table}) {
                        $fpre =
@load_flatfile_area_list[$i].$params{'load_dbgen_'.$table.'_output_
prefix'};
                }
                else {
                        $fpre =
$params{'load_flatfile_area'}.$params{'load_dbgen_'.$table.'_output
_prefix'};
          }
                if ($dgp)
                {
                        # create script to generate partitioned flat files

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

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


sub dbgen_old
{

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

    $cur_inst = 1 if $multi;  # for possible locality on SP2

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

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

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

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

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

```
        }

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

        # Start - Barry N. Perkins 8/27/1998          #
        # included the following lines to copy flat files#
        # from default location to backup location      #
        # Defined a new variable for backup location    #
        $fbkp = (defined
$params{'load_dbgen_flatfile_backup'}) ?
            $params{'load_dbgen_flatfile_backup'} :
            0;
        $bpre = (defined
$params{'load_dbgen_backup_'.$table.'_prefix'}) ?
            $params{'load_dbgen_backup_'.$table.'_prefix'} :
           $params{'load_dbgen_backup_area'} . $table;
        # End   - Barry N. Perkins 8/27/1998          #
        # if using pipes, create the pipes
        if($dgp)
        {
           # create script to generate partitioned flat files for each
dbgpar

            $nperset = 1;

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

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

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

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

sub dbgen_nt
{

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

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

```
	$bfexist = 0;
	$dafexist = 0;
	$difexist = 0;
	$ffexist = 0;
		$up = 0;
		$dgp = 0;
		$dbgpar = 1;

		# see if we are using pipes

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


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


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

		if ($up) {
		  $dbgopt .= " -T ".$dbgtab." ";
		  $dbgopt .= " -s ". $params{'scale_factor'} . " ";
		  $dbgopt .= " -C ";
		  $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?

		$params{'load_dbgen_'.$table.'_option_C'} :

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

		$params{'load_dbgen_'.$table.'_option_C'} :

		$params{'load_dbgen_def_option_C'});
		}

		# if using pipes, create the pipes

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

		  $nperset = 1;

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

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

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

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

            &dump0("*wait");

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

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


sub dbgen_unix
{

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

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

          # see if we are using pipes

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

      $params{'tab_'.$table.'_load_ctlf'} = $table.'.ctl' if !defined
$params{'tab_'.$table.'_load_ctlf'};
      @ctlf = split(/,/, $params{'tab_'.$table.'_load_ctlf'});
      $err = "ctl" if ((@ctlf > 1) && (@ctlf != $dp));
      @logf = split(/,/, $params{'tab_'.$table.'_load_logf'});
      $err = "log" if ((@logf > 1) && (@logf != $dp));
      $lfexist = 1 if @logf > 0;
      @badf = split(/,/, $params{'tab_'.$table.'_load_badf'});
      $err = "bad" if ((@badf > 1) && (@badf != $dp));
      $bfexist = 1 if @badf > 0;
      @datf = split(/,/, $params{'tab_'.$table.'_load_datf'});
      $err = "dat" if ((@datf > 1) && (@datf != $dp));
      $dafexist = 1 if @datf > 0;
      @disf = split(/,/, $params{'tab_'.$table.'_load_disf'});
      $err = "dis" if ((@disf > 1) && (@disf != $dp));
      $difexist = 1 if @diff > 0;
          @filf = ();
#       $params{'tab_'.$table.'_load_filf'} =
$params{$curts.'_datafiles'} if ($params{'tab_'.$table.'_load_filf'}
=~ /alltsdatafiles/);
      if ($params{'tab_'.$table.'_load_filf'} =~ /alltsdatafiles/)
      {
       $params{'tab_'.$table.'_load_filf'} =
$params{$curts.'_datafiles'}
      }
      @filf = split(/,/, $params{'tab_'.$table.'_load_filf'}) if defined
$params{'tab_'.$table.'_load_filf'};
          # relax the requirement a little bit by allowing round
robining
          $numfil = @filf;
#         if (($numfil > 1) && ($numfil < $dp) && (($dp %
$numfil) == 0)) {
     if (($numfil > 1) && ($numfil < $dp)) {
#           $p = $dp / $numfil;
            $fil = "";
            for ($i = 0; $i < $dp; $i++)
            {
                  if ($i == 0) {
                     $fil = $params{'tab_'.$table.'_load_filf'};
                  } else {
                     $fil =
join(',',$fil,$params{'tab_'.$table.'_load_filf'});
                  }
            }
            @filf = split(/,/, $fil);
          }
      $err = "fil" if ((@filf > 1) && ($dp != @filf));
      $ffexist = 1 if @filf > 0;
      if ($err)
      {
```

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

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

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

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

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

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

      $ff_path = $params{'load_dbgen_'.$table.'_output_prefix'};
      $numchild = $params{'load_dbgen_'.$table.'_option_C'};
      $pname = $params{'tab_'.$table.'_partnames'};
      @pname = split(/,/, $pname);
# create the controlfiles (fixed-length fields or delimited records)
#       if ($params{'skip_mk_ldctlfs'} !~ /$table/)
#       {
         $npart = (defined $params{'tab_'.$table.'_#part'}) ?
$params{'tab_'.$table.'_#part'} : 1;
         for ($ictl=1; $ictl<=$npart; $ictl++) {
           open (CTLFILE, ">$control");
           print CTLFILE "---\n";
           print CTLFILE "---  $table.ctl for delimited records\n" if
($ud == 1);
```

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

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

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

                $params{'load_dbgen_'.$table.'_option_C'} :

                $params{'load_dbgen_def_option_C'});
                if ($dgp) {
                  $dbgopt .= " -p -i ";
                  $dbgopt .= (defined
$params{'load_dbgen_'.$table.'_input_params'}) ?
$params{'load_dbgen_'.$table.'_input_params'} :
$params{'load_dbgen_def_input_params'};
                }
              }

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

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

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

          # if using pipes, create the pipes

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

                # start the dbgens

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

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

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

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

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

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

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

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


sub plcre
{

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

# create NT Raw Partition and Setlink input files
    $lnksfile = $params{'plcre_setlinks_input_file'};
    open (LNKSFILE, ">$lnksfile");
    $drivenum = $params{'plcre_drivenums'};
    $md = $drivenum =~ tr/,/,/;
    @drivenum = split(/,/,$params{'plcre_drivenums'});
if ($params{'plcre_recreate_extended_partitions' =~
/[tT][rR][uU][eE]/})
    {
             foreach $drivenum (@drivenum)
             {
                &recreate_drive_extended_part;
             }
             if (defined $params{'plcre_log_drivenum'})
             {
```

```
                &recreate_drive_extended_part;
            }
    }
    $io_control_files = $params{'io_control_files'};
    $io_control_files =~ s/[\(\"\\\.\)]*//g;
    @io_control_files = split(/,/,$io_control_files);
    $d = 0;
    foreach $file (@io_control_files)
    {
            $size = $params{'plcre_control_file_size'};
            $size =~ s/[Mm]*//g;
            &create_drive_part_file;
    }
    foreach $ts_entry (@ts_all)
    {
            &create_drive_part("$ts_entry") if ($ts_entry =~ /ts_sys/
|| $ts_entry =~ /ts_log/);
    }
    $d = 0;
    foreach $ts_entry (@ts_most)
    {
            &create_drive_part("$ts_entry");
    }
    close (LNKSFILE);
    &dump0("*wait");
    &time0("Done NT Raw Partition creation");
    $cmd = "setlinks /F:" . $lnksfile . "\n";
    &dump1("*sh");
    &dump0("*wait");
    &time0("Done Setlink links creation");
    &time0("Done NT Raw Partition and Setlink creation");
} # end of plcre




sub sccre
{

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

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

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

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

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

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

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


sub sctso
{

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

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

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

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

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


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

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

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

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

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


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

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

               if (defined(@tablelog_list))
               {

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

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

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

               foreach $view (@view_list)
               {
                  @viewlogs =
split(/,/,$params{'view_'.$view.'_viewlogs'});
```

```
                foreach $viewlog (@viewlogs)
                {
                        $ob_type = 'viewlog';
                        $object = $viewlog;
                        if (!(defined $params{'created_'.$object}) )
                        {
                           $params{'created_'.$object} = $object;
                           &create_object;
                        }
                }
                $ob_type = 'view';
                $object = $view;
                &create_object;
             }
             &dump1("*sql");




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

       &dump0("*wait")
               }
}

sub dsffs
{

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

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

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

}
sub dapop # Meikel's version
{

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

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


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

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

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

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

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

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

      $ff_path =
$params{'load_flatfile_area'}.$params{'load_dbgen_'.$table.'_output
_prefix'};
               $pname = $params{'tab_'.$table.'_partnames'};
               $pname =~ s/\#//g;
               print $pname."--".@pname;
               @pname = split(/\,/,$pname);
               $npart = ((defined $params{'tab_'.$table.'_#part'}) &&
$dgp) ?
                  $params{'tab_'.$table.'_#part'} : 1;
               $ctlarea = $params{'load_controlfile_area'};
               $logarea = $params{'load_otherfile_area'};

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

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

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

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

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

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

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

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

# create the controlfiles (fixed-length fields or delimited records)
               if ($params{'skip_mk_ldctlfs'} !~ /$table/)
               {
                     &load_ctl_head;
                     $infiles = 0;
                     $badfiles = 0;
                     $discardfiles = 0;
                     $intopartitions = "BEGINNING"; # this will
collect the partition numbers that will be served.
                     if ($dp < $npart) {
                        if ($idp <= $remchild) {
                              $nchild = $numchild+1;
                        }else{
                              $nchild = $numchild;
                        }
                     }else{
                        if ($partcount <=
$sqlldr_per_partition_rest) {
                              $nchild = $infiles_per_sqlldr+1;
                        }else{
                              $nchild = $infiles_per_sqlldr;
                        }
                     }
                     for ($jpart=1; $jpart<=$nchild; $jpart++)
                     {
                        #print "ipart $ipart \n";
                        $idbgpar = 0 if $idbgpar >= $dbgpar;
                        #$ipart = int((($idp -
1)*$nchild+$jpart)/$dbgpar);
                        #$ipart = $ipart+1 if ((((($idp-
1)*$nchild+$jpart)-$ipart*$dbgpar)>0);
                        if ($dgp)
                        {
```

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

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


sub sqlldr_ctl_ops
{
   $controlfiles_per_node = $dpmax / $nnodes;
   print "controlfiles_per_node $controlfiles_per_node nnodes
$nnodes\n";
   for ($i=1; $i<=$nnodes; $i++){
          @ipart[$i] = (($i-1)*$npartitions_per_node)+1;
          @iipart[$i] = 1;
          print "Offset for node $i is: @ipart[$i]\n";
   }
```

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

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

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

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

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



sub dapop_frank ## copied from Frank's version
{

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

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

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

                # see if we are using pipes

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

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

## NT Port
## exapnd all files for all partitions
                if ($params{'tab_'.$table.'_load_filf'} =~ /alltsdatafiles/ )
```

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

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

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

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

## NT End

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

```
                        $fil =
join(',',$fil,$params{'tab_'.$table.'_load_filf'});
                    }
                }
                @filf = split(/,/, $fil);
            }

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

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

    @ff = ();

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

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

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

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

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

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

		# add code to create head and tail of control files

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

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

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

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

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


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

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

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

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

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

		if ($up) {
		   if ($table =~ /lineitem/)
		   {
			$dbgtab = "L";
		   }
		   elsif ($table =~ /orders/)
		   {
			$dbgtab = "O";
		   }
		   elsif ($table =~ /partsupp/)
		   {
			$dbgtab = "S";
		   }
```

```
                elsif ($table =~ /part/)
                {
                        $dbgtab = "P";
                }
                elsif ($table =~ /customer/)
                {
                        $dbgtab = "c";
                }
                elsif ($table =~ /supplier/)
                {
                        $dbgtab = "s";
                }
                elsif ($table =~ /nation/)
                {
                        $dbgtab = "n";
                }
                elsif ($table =~ /region/)
                {
                        $dbgtab = "r";
                }
                $dbgopt .= " -T ".$dbgtab." ";
                $dbgopt .= " -s ". $params{'scale_factor'} . " ";
                $dbgopt .= " -C ";
                $dbgopt .= ((defined
$params{'load_dbgen_'.$table.'_option_C'}) ?

                $params{'load_dbgen_'.$table.'_option_C'} :

                $params{'load_dbgen_def_option_C'});
                   if ($dgp) {
                        $dbgopt .= " -p -i ";
                        $dbgopt .= (defined
$params{'load_dbgen_'.$table.'_input_params'}) ?
$params{'load_dbgen_'.$table.'_input_params'} :
$params{'load_dbgen_def_input_params'};
                   }
                }

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

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

## NT Port (use dbgen to create flat files)
##
##              # if using pipes, create the pipes
##
##              if($up)
##              {
##                      for ($i=0;$i<$dp;$i++) {
##                              $cmd = sprintf("%s%s
p\n",$params{'load_flatfile_area'},@daf[$i]);
##                              &dump1("*mknod");
##                      }
##                      &dump0("*wait");
##
##                      # start the dbgens
##
##                      if ($dgp) {
##                              $nset = int($dp /
$params{'tab_'.$table.'_#part'});
##                              $nperset = int($dbgpar / $nset);
##                      }
##
##                      for ($i=0;$i<$dbgpar;$i++) {
##                              $cmd = $params{'load_flatfile_area'} . "\n";
##                              $cmd2 = sprintf("%s -S %d", $dbgopt, ($i +
1));
##                              if ($dgp) {
##                                 # if the output prefix name is specified with
#
##                                 # divide output file names into
$dp/#numpart sets
##                                 $fpre = (defined $params{'load_dbgen_'.
$table.'_output_prefix'}) ?
$params{'load_dbgen_'.$table.'_output_prefix'} :
$params{'load_dbgen_def_output_prefix'};
##                                 if ($fpre =~ /\#/) {
##                                         $snum = int($i/$nperset) + 1;
##                                         $fpre =~ s/\#/$snum/g;
##                                 }
##                                 $cmd2 .= sprintf(" -o %s ", $fpre);
##                              }
##
### Added 4/28? - Ari
### need to force in case pipe still there, and need to kick
### off in the backgroud from here
##                              $cmd2 = " -f " . $cmd2;
##                              $cmd2 .= " & ";
##
##                              $cmd2 .= "\n";
##                              &dump2("*dbgen");
##                      }
##              }
##
##              &dump0("*wait");
## NT End

        for ($i=0; $i < $dp; $i++)
        {
           &advmulti();
                $control = "";
                $log = "";
                $bad = "";
                $data="";
                $discard = "";
           $control = sprintf (" control=%s%s",
$params{'load_controlfile_area'}, @cf[$i]);
           $control = sprintf (" control=/host%s%s",
$params{'load_controlfile_area'}, @cf[$i]) if
($params{'special_machine'} eq 'ncube');
           $log = sprintf (" log=%s%s", $params{'load_otherfile_area'},
@lf[$i]) if ($lfexist == 1);
           $bad = sprintf (" bad=%s%s",
$params{'load_otherfile_area'}, @bf[$i]) if ($bfexist == 1);
           $data = sprintf (" data=%s%s", $params{'load_flatfile_area'},
@daf[$i]) if ($dafexist== 1);
           $discard = sprintf (" discard=%s%s",
$params{'load_otherfile_area'}, @dif[$i]) if ($difexist== 1);
```

```
                if ($ffexist ==1)
                {
            if ($curts == "")
                {
                        $file = sprintf (" file=%s" , @ff[$i]);
                    }
                    else
                    {
            $file = sprintf (" file=%s%s", $params{$curts.'_area'},
@ff[$i]);
                    }
                }
## NT Port
## Change to UPPER case string as a workaround for SQLLDR

                $file =~ tr/a-z/A-Z/;
## NT End
        if ($usels)
        {
           $loadval = $loadval + 1 if ($extra > $i);
            $load = sprintf (" load=%d", $loadval);
           $skip = sprintf (" skip=%d", $skipval);
           $skipval = $skipval + $loadval;
           $loadval = $split;
        }
##              if ($up)
##              {
##                      $load = " load=99999999";
##              }

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

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

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



sub ixcre
{

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

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

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

```
                        $cmd .= " (degree
$params{'ind_'.$index.'_pardeg'} "
                            if defined $params{'ind_'.$index.'_pardeg'};
                        $cmd .= "instances
$params{'ind_'.$index.'_parinst'}"
                            if defined $params{'ind_'.$index.'_parinst'};
                        $cmd = $cmd . ")\n";
                    }
                }

                # deal with partitioning

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


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

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

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

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

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

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

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

   &process_ind_part_ts;

   # complete the local partition index statement
   if ($numpart >0)
   {
            $cmd .= "(\n";
            for ($i=0; $i < $numpart; $i++)
            {
               $cmd .= "partition ";
               $cmd .= $ind_partnames[$i] . "\n" if @ind_partnames;
               &process_part_params('ind','%f','pctfree',$index,
                                                   (@ind_partnames) ?
$ind_partnames[$i] : "");
               &process_part_params('ind','%u','pctused',$index,
                                                   (@ind_partnames) ?
$ind_partnames[$i] : "");
               &process_part_params('ind','it','initrans',$index,
                                                   (@ind_partnames) ?
$ind_partnames[$i] : "");
               &process_part_params('ind','mt','maxtrans',$index,
                                                   (@ind_partnames) ?
$ind_partnames[$i] : "");
               if ((@ind_partnames) &&
                        (defined
$params{'ind_'.$index.'_'.$ind_partnames[$i].'_ts'}))
               {
                        $cmd .= 'tablespace '.

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

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

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


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

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

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

   if ($ind_partnames[0] =~ /\#/)
   {
            $filenm = $ind_partnames[0];
            $savename = $filenm;
```

```
            $params{'ind_'.$index.'_partnames'} = "";
            # indtab defined in ixcre
            for ($i = 1; $i <= $numpart; $i++)
            {
               $filenm =~ s/\#/$i/g;
               ($i == $numpart) ? ($addcomma = "") : ($addcomma =
",");
               $params{'ind_'.$index.'_partnames'} =
                        $params{'ind_'.$index.'_part names'} . $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_par
tnames[$i]);
            &process_part_params('ind','%u','pctused',$index,$ind_p
artnames[$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].'_nolg'})
            {
               $cmd .= "nologging\n"
                        if
($params{'ind_'.$index.'_'.$ind_partnames[$i].'_nolg'}
                           =~ /[tT][rR][uU][eE]/);
            } elsif (defined $params{'ind_'.$table.'_part_def_nolg'})
{
               $cmd .= "nologging\n"
                        if ($params{'ind_'.$index.'_part_def_nolg'}
=~
                           /[tT][rR][uU][eE]/);
            }
            $cmd .= (($i+1) == $numpart) ? "" : ",\n";
   }
   $cmd .= ")\n";

} # end of exp_ind_part_g


sub process_ind_part_ts
{
   # is ind_<name>_part_ts is in the form of XXXX#, expand it
   # else treat it as a comma separated list of ts names
   if (defined $params{'ind_'.$index.'_part_ts'})
   {
      @ind_part_ts = split(/,/,$params{'ind_'.$index.'_part_ts'});
      if ($ind_part_ts[0] =~ /\#/)
      {
         $filenm = shift(@ind_part_ts);
         $savename = $filenm;
         $params{'ind_'.$index.'_part_ts'} = "";
         for ($i = 1; $i <= $numpart; $i++)
         {
            $filenm =~ s/\#/$i/g;
            ($i == $numpart) ? ($addcomma = "") : ($addcomma =
",");
            $params{'ind_'.$index.'_part_ts'} =
               $params{'ind_'.$index.'_part_ts'} . $filenm .
$addcomma;
            $filenm = $savename;
         }
         @ind_part_ts = split(/,/,$params{'ind_'.$index.'_part_ts'});
      } else {
         if (@ind_part_ts != $numpart)
         {
                        $numpart = @ind_part_ts;
                        if (($numpart % $numfil) == 0) {
                           $p = $numpart / $num fil;
                           $fil = "";
                           for ($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 ($col =~ /orderkey$/)
      {
                $high_l_orderkey = $params{'scale_factor'} * 1500000
* 4 if ($col =~ /^l_orderkey$/);
                $high_l_orderkey = $params{'scale_factor'} * 1500000
* 4 if ($col =~ /^o_orderkey$/);
                $pval = 1;
       if (defined $params{$ob_stype.'_'.$table.'_#part'})
       {
        $numpart = $params{$ob_stype.'_'.$table.'_#part'};
       }
       else
       {
        $numpart = $params{'ind_'.$index.'_#part'};
       }
              $rest = $high_l_orderkey % $numpart;
       $incre = ($high_l_orderkey-$rest) / $numpart;
       for ($i=0; $i < $numpart-1; $i++) {
         $pval = $pval + $incre;
         push(@ind_part_col, $pval);
       }
       push(@ind_part_col, 'MAXVALUE');
      }
      else {
       if ($col =~ /^l_partkey$/) {
        $high_l_partkey = $params{'scale_factor'} * 200000;
        $pval = 1;
        if (defined $params{$ob_stype.'_'.$table.'_#part'})
        {
         $numpart = $params{$ob_stype.'_'.$table.'_#part'};
        }
        else
        {
         $numpart = $params{'ind_'.$index.'_#part'};
        }
        $incre = $high_l_partkey / $numpart;
        for ($i=0; $i < $numpart-1; $i++) {
          $pval = $pval + $incre;
          push(@ind_part_col, $pval);
        }
       push(@ind_part_col, 'MAXVALUE');
       }
       else
       {
        @ind_part_col =
split(/,/,$params{'ind_'.$index.'_'.$col.'_partvals'});
       }
      }

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

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


sub anlyz
{

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

   if ($params{'analyze_type'} =~ /via package dbms.stats/)
   {
            &time0("Begin via package dbms.stats analyzing");
            foreach $object (@anlyzlist)
            {
               $cmd = "connect
$params{'user'}/$params{'passwd'};\n";
               $phead="anl_".$object;
```

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

$params{$tab_entry.'_partnames'} . $nextfile;
                  }
                }
                @tab_partnames =
split(/,/,$params{$tab_entry.'_partnames'});
                # if partnames is specified as XXXX#, then
expand
                if ($tab_partnames[0] =~ /#/)
                {
                  $filenm = shift(@tab_partnames);
                  $savename = $filenm;
                  $params{$tab_entry.'_partnames'} = "";
                  for ($i = 1; $i <=
$params{$tab_entry.'_#part'}; $i++)
                  {
                        $filenm =~ s/\#/$i/g;
                        ($i ==
$params{$tab_entry.'_#part'}) ? ($addcomma = "") :
                            ($addcomma = ",");
                        $params{$tab_entry.'_partnames'}
=

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

$params{'tab_'.$params{$ind_entry.'_table'}.'_#part'};
                  }
                  if (!defined
$params{$ind_entry.'_partnames'})
                  {
                        # if no partnames are specified, use
a default part name
                        for ($i = 1; $i <=
$params{$ind_entry.'_#part'}; $i++)
                        {
                          ($i ==
$params{$ind_entry.'_#part'}) ?
```

```
                                        ($addcomma = "") :
($addcomma = ",");
                                    $nextfile =
sprintf("%s%s%d%s",$object,"_p",

        $i, $addcomma );

$params{$ind_entry.'_partnames'} =

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

sub chdop_old
{

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

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

sub chob
{
```

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

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



sub expln
{

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

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

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

sub query
{

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

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

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

sub prepmulti
{
# Some multi-instance stuff
  $num_inst = $params{$phase.'_num_inst'};
  if ($num_inst > 1)
  {
    $multi = 1;
    $cur_inst = 1;
    &advmulti(); # Make sure it's set to some appropriate value
```

```
    }
    else
    {
       $multi = 0;
    }
}

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

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

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

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

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

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

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

```
                        }
                        else
                        {
                           $cmd .= "uniform";
                        }
                     }
                  else                        # system managed permanent
tablespace
                  {
                     $cmd .= "drop tablespace $tsname including
contents;\n";
                     $cmd .= "create tablespace $tsname\n";
                     $cmd .= "datafile
'$params{$tsname.'_area'}$ts_datafile' size
$params{$tsname.'_first_size'} $params{$tsname.'_options'}\n";
                     $cmd .= "extent management local\n";
                     if (defined ($params{$tsname.'_extent_size'}) &&
($params{$tsname.'_extent_size'} =~
/[aA][uU][Tt][oO][Aa][lL][lL][oO][cC][aA][tT][eE]/))
                     {
                              $cmd .= "autoallocate\n";
                     }
                     elsif (defined ($params{$tsname.'_extent_size'}) &&
!($params{$tsname.'_extent_size'} =~
/[aA][uU][Tt][oO][Aa][lL][lL][oO][cC][aA][tT][eE]/))
                     {
                              $cmd .= "uniform size
".$params{$tsname.'_extent_size'}."\n";
                     }
                     else
                     {
                              $cmd .= "uniform";
                     }
                  }
   }
   $cmd .= " nologging\n" if ($params{$tsname.'_nolg'} =~
/[tT][rR][uU][eE]/);
   $cmd .= ";\n";
   &dump1("*sql");
}

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

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

}

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

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

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

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


sub create_objects

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


sub create_object

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

{

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

```
                $cmd .= "initrans $params{$ob_entry.'_of_it'} " if
defined $params{$ob_entry.'_of_it'};
                $cmd .= "maxtrans $params{$ob_entry.'_of_mt'} " if
defined $params{$ob_entry.'_of_mt'};
                $cmd .= "tablespace $params{$ob_entry.'_of_ts'} " if
defined $params{$ob_entry.'_of_ts'};
                $cmd .= "storage $params{$ob_entry.'_of_storage'}\n"
if defined $params{$ob_entry.'_of_storage'};
            }
    }
    if ((defined $params{$ob_entry.'_pardeg'}) || (defined
$params{$ob_entry.'_parinst'}))
    {
            $cmd .= "parallel";
            if ((defined $params{$ob_entry.'_pardeg'}) &&
                ($params{$ob_entry.'_pardeg'} =~
/[dD][eE][fF][aA][uU][lL][tT]/))
            {
                $cmd .= "\n";
            }
            else
            {
                $cmd .= "(degree $params{$ob_entry.'_pardeg'} " if
defined $params{$ob_entry.'_pardeg'};
                $cmd .= "instances $params{$ob_entry.'_parinst'}" if
defined $params{$ob_entry.'_parinst'};
                $cmd .= ")\n";
            }
    }
    $cmd .= "nologging\n" if ($params{$ob_entry.'_nolg'} =~
/[tT][rR][uU][eE]/);
    $cmd .= "cache\n" if ($params{$ob_entry.'_cache'} =~
/[tT][rR][uU][eE]/);

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


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

            if ($params{$ob_entry.'_rewrite'} =~ /[tT][rR][uU][eE]/)

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

    if (defined $params{$ob_entry.'_as_select'}) # as select (here I
only copy a variable called <viewname>_define_as_select
    {
            &print_as_select_stm($params{$ob_entry.'_as_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/(, )|(,\t)/,\n /g;
   $as_select =~ s/from|FROM/\nFROM\n/g;
   $as_select =~ s/select|SELECT/SELECT \n/g;
   $as_select =~ s/where|WHERE/\nWHERE\n/g;
   $as_select =~ s/group by|GROUP BY/\nGROUP BY\n/g;
   $as_select =~ s/ and | AND /\n AND \n/g;
   $as_select =~ s/ or | OR /\n OR \n/g;
   $cmd .= "as select \n";
   $cmd .= "/*".$hint."*/"."\n" if ($hint !~ /\^/);
   $cmd .= $as_select;
}
sub create_clusters
{
   $cmd .= "connect $params{'user'}/$params{'passwd'};\n";
   foreach $cluster (@clulist)
   {
      $cmd .= "drop cluster $cluster including tables;\n";
      $cmd .= "create cluster $cluster (\n";
      @clu_collist = split(/,/, $params{'clu_'.$cluster.'_columns'});
      while ($col = shift(@clu_collist))
      {
         (@clu_collist == 0) ? ($addcomma = "") : ($addcomma =
",");
        $nextline = sprintf ("    %-20s %s$addcomma\n", $col,
$params{'clu_'.$cluster.'_'.$col.'_type'});
        $cmd .= $nextline;
      }
      $cmd .= ")\n";
      $cmd .= "pctfree $params{'clu_'.$cluster.'_%f'}\n" if defined
$params{'clu_'.$cluster.'_%f'};
      $cmd .= "pctused $params{'clu_'.$cluster.'_%u'}\n" if defined
$params{'clu_'.$cluster.'_%u'};
      $cmd .= "initrans $params{'clu_'.$cluster.'_it'}\n" if defined
$params{'clu_'.$cluster.'_it'};
      $cmd .= "maxtrans $params{'clu_'.$cluster.'_mt'}\n" if defined
$params{'clu_'.$cluster.'_mt'};
      $cmd .= "size $params{'clu_'.$cluster.'_size'}\n" if defined
$params{'clu_'.$cluster.'_size'};
      $cmd .= "tablespace $params{'clu_'.$cluster.'_ts'}\n" if defined
$params{'clu_'.$cluster.'_ts'};
      $cmd .= "storage $params{'clu_'.$cluster.'_storage'}\n" if
defined $params{'clu_'.$cluster.'_storage'};
      $cmd .= "index\n" if ($params{'clu_'.$cluster.'_index'} =~
true);
      if (defined $params{'clu_'.$cluster.'_hashkeys'})
      {
         $cmd .= "hash is $params{'clu_'.$cluster.'_hashcol'} " if
defined $params{'clu_'.$cluster.'_hashcol'};
         $cmd .= "hashkeys $params{'clu_'.$cluster.'_hashkeys'}\n";
      }
      if ((defined $params{'clu_'.$cluster.'_pardeg'}) || (defined
$params{'clu_'.$cluster.'_parinst'}))
      {
         $cmd .= "parallel ";
         $cmd .= "degree $params{'clu_'.$cluster.'_pardeg'} " if
defined $params{'clu_'.$cluster.'_pardeg'};
         $cmd .= "instances $params{'clu_'.$cluster.'_parinst'}" if
defined $params{'clu_'.$cluster.'_parinst'};
         $cmd .= "\n";
      }
      $cmd .= "cache\n" if $params{'clu_'.$cluster.'_cache'} =~
/true/;
      $cmd .= ";\n";
# Now, create the cluster index, if necessary
      if ($params{'clu_'.$cluster.'_index'} =~ true)
      {
         $cluindex = "ind_".$cluster;
         $cmd .= "drop index $cluindex;\n";
         $cmd .= "create index $cluindex on cluster $cluster;\n";
      }
   }
   &dump1("*sql");
}


sub expand_partitions

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

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


   if ($params{$ob_entry.'_parttype'} =~ /[rR][aA][nN][gG][eE]/)
   {
        $cmd .= "partition by range (" .
$params{$ob_entry.'_partcol'} . ")\n(\n";
        &expand_partitions_doit($ob_entry,$ob,$ob_type);
   }
   elsif ($params{$ob_entry.'_parttype'} =~ /[hH][aA][sS][hH]/)
   {
        $cmd .= "partition by hash (" .
$params{$ob_entry.'_partcol'} . ")\n";
        $cmd .= "partitions ". $params{$ob_entry.'_#part'} . "\n";
   }
   elsif ($params{$ob_entry.'_parttype'} =~
/[cC][oO][mM][pP][oO][sS][iI][tT][eE]/)
   {
```

```
            $cmd .= "partition by range (" .
$params{$ob_entry.'_partcol'} . ")\n";
            $cmd .= "subpartition by hash(".
$params{$ob_entry.'_subpartcol'} . ")\n";
            $cmd .= "subpartitions ".
$params{$ob_entry.'_#subpart'} . "\n(\n";
            &expand_part itions_doit($ob_entry,$ob,$ob_type);
    }
}                                   # no partitioning


sub expand_partitions_doit

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

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

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

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

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

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

    # complete the partition statement

    for ($i=0; $i < $params{$ob_entry.'_#part'}; $i++)
    {
            $cmd .= "partition " . $ob_partnames[$i] . " values less
than ";
       if ($i==$params{$ob_entry.'_#part'}-1) {
         $cmd .= "(MAXVALUE)\n";
       }
       else {
            $cmd .= "(" . $ob_part_vals[$i] . ")\n";
       }
            &process_part_params($ob_type,'%f','pctfree',$ob,$ob_p
artnames[$i]);
            &process_part_params($ob_type,'it','initrans',$ob,$ob_pa
rtnames[$i]);
            &process_part_params($ob_type,'mt','maxtrans',$ob,$ob_
partnames[$i]);

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

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

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


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


sub process_part_params
{
```

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


sub process_part_ts

# *MP* parameterize procedure: changed 't ab'.$table into
$ob__entry 1. parameter $_[0] (same as $_entry in expand_partition)
#                               $table into $ob_          2. parameter $_[1]
(same as $ob in expand_partition)

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


sub process_part_brys


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

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

    $cnt = 0;
    @ob_part_vals = ();

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


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

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

```
            } else {
                        $nls_format = (defined
$params{$ob__entry.'_'.$col.'_date_format'}) ?
$params{$ob__entry.'_'.$col.'_date_format'} : "YYYY-MM-DD";
                        $ob_part_vals[$i] .=
"to_date('".$ob_part_col[$i]."','";
                        $ob_part_vals[$i] .=
$nls_format."')".$addcomma;
            }
                } else {
                        $ob_part_vals[$i] = $ob_part_vals[$i] .
$addquote.
                          $ob_part_col[$i] . $addquote . $addcomma
;
                }
             }
             $cnt++;
    }
} # end of process_part_brys

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

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

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

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

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

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

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

}

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

sub load_ctl_head
{
   print "control $control\n";
   open (CTLFILE, ">$control");
   print CTLFILE "---\n";
```

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

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

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

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


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

$params{'dd_sqlplus_area'}="$ENV{'ORACLE_HOME'}/sqlplus/a
dmin/" if !(defined $params{'dd_sqlplus_area'});

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

   #------------------------ undo tablespace definition ----------------------
---------------
   $params{'ts_undo_#rs'} = $params{'def_num_inst'} if !defined
$params{'ts_undo_#rs'};
   $params{'ts_undo_area'} = $params{'dbs_area'} if !defined
$params{'ts_undo_area'};
   $params{'ts_undo_rs_prefix'} = 'r' if !defined
$params{'ts_undo_rs_prefix'};
   $save_datafiles = $params{'ts_undo_datafiles'};

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

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

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

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

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

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

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

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

               $pr = $params{'ts_undo_rs_prefix'}.$node;
```

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

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

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

    # expand ts_data groups

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

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

# expand tablespaces
    @ts_data_new =();
    $ts_data_names="@@";
    foreach $ts_entry (@ts_data)
    {
            if (defined($params{$ts_entry.'_instances'})){
               for ($i = $params{$ts_entry.'_instances'}; $i > 0; $i --)
{
                        $ts_new_entry = $ts_entry.$i;

            $params{$ts_new_entry.'_size'}=$params{$ts_entry.'_siz
e'};

            $params{$ts_new_entry.'_datafiles'}=$params{$ts_entry.
'_datafiles'};

            $params{$ts_new_entry.'_datafiles'}=~s/\#/$i/;

            $params{$ts_new_entry.'_#files'}=$params{$ts_entry.'_#
files'};

            $params{$ts_new_entry.'_storage'}=$params{$ts_entry.'
_storage'};
                        unshift (@ts_data_new, $ts_new_entry);

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

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

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


    #rint "ts_all: @ts_all\n";
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

   @indexlist = split(/,/, $params{'ind_indices'});
   foreach $index (@indexlist)
   {
      $params{'ind_'.$index.'_ts'} = "ts_index" if ((!defined
$params{'ind_'.$index.'_ts'}) && ($params{'skip_ts'} !~ /index/));
   }
   $params{'analyze_type'} = "via package dbms.stats" if
!defined($params{'analyze_type'});
   @anlyzlist = split(/,/, $params{'anl_objects'});
   @anlyzlist = "schema" if (@anlyzlist == 0);
```

```
    foreach $object (@anlyzlist)
    {
            $params{'anl_'.$object.'_object_type'} = "schema" if
($object =~ /schema/);
       $params{'anl_'.$object.'_object_type'} = "table" if !defined
$params{'anl_'.$object.'_object_type'};
       $params{'anl_'.$object.'_anl_type'} = "gather" if !defined
$params{'anl_'.$object.'_anl_type'};
    }
    @choblist = split(/,/, $params{'chob_objects'});
#    foreach $object (@choblist)
#    {
#      $choblist{$object} = $params{'chob_'.$object.'_pardeg'};
#    }
    @chstorlist = split(/,/, $params{'chstor_objects'});
    foreach $object (@chstorlist)
    {
      $chstorlist{$object} = $params{'chstor_'.$object.'_next'};
    }
    if ((defined $params{'verbose'}) && (($params{'verbose'} =~
/[tT][rR][uU][eE]/) || ($params{'verbose'} =~ 1))){
            $verbose=1;}
    else {
            $verbose=0;}
    if ((defined $params{'log_output'}) && (($params{'log_output'}
=~ /[tT][rR][uU][eE]/) || ($params{'log_output'} =~ 1))){
            $log_output=1;}
    else {
            $log_output=0;}
    $params{'plcre_recreate_extended_partitions'} = "false" if
!defined ($params{'plcre_recreate_extended_partitions'});
}

sub assigndefs
{
    while ($onedef = shift(@defparams))
    {
       $onedef =~ /^(.*)=(.*)$/;
       $key = $1;
       $value = $2;
       $params{$key} = $value if !defined $params{$key};
    }
}

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

sub tpcd
{
    @defparams = ();
    push (@defparams, 'user=tpcd');
    push (@defparams, 'passwd=tpcd');
    push (@defparams,
'tab_tables=lineitem,orders,partsupp,part,customer,supplier,nation,re
gion');
    push (@defparams, 'view_views=mav_li,mv_locs');
    push (@defparams, 'viewlog_list=lineitem,orders');
    push (@defparams,
'mav_li_columns=count_p_group,l_shipdate,l_returnflag,l_linestatus
,sum_qty,sum_base_price ,sum_disc_price
,count_disc_price,sum_charge,
count_charge,count_qty,count_price,sum_disc,count_disc,count_ord
er');
    push (@defparams,
'tab_lineitem_columns=l_orderkey,l_partkey,l_suppkey,l_linenumbe
r,l_quantity,l_extendedprice,l_discount,l_tax,l_returnflag,l_linestatus
,l_shipdate,l_commitdate,l_receiptdate,l_shipinstruct,l_shipmode,l_c
omment');
    push (@defparams,
'tab_orders_columns=o_orderkey,o_custkey,o_orderstatus,o_totalpri
ce,o_orderdate,o_orderpriority,o_clerk,o_shippriority,o_comment');
    push (@defparams,
'tab_partsupp_columns=ps_partkey,ps_suppkey,ps_availqty,ps_supp
lycost,ps_comment');
    push (@defparams,
'tab_part_columns=p_partkey,p_name,p_mfgr,p_brand,p_type,p_siz
e,p_container,p_retailprice,p_comment');
    push (@defparams,
'tab_customer_columns=c_custkey,c_name,c_address,c_nationkey,c
_phone,c_acctbal,c_mktsegment,c_comment');
    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_regionkey,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_part_#rows=200000');
    push (@defparams, 'tab_customer_#rows=150000');
    push (@defparams, 'tab_supplier_#rows=10000');
    push (@defparams, 'tab_nation_#rows=0');
    push (@defparams, 'tab_region_#rows=0');
    push (@defparams, 'tab_lineitem_parttype=range');
    push (@defparams, 'tab_lineitem_l_orderkey_type=number');
    push (@defparams, 'tab_lineitem_l_partkey_type=number');
    push (@defparams, 'tab_lineitem_l_suppkey_type=number');
    push (@defparams, 'tab_lineitem_l_linenumber_type=number');
    push (@defparams, 'tab_lineitem_l_quantity_type=number');
    push (@defparams,
'tab_lineitem_l_extendedprice_type=number');
    push (@defparams, 'tab_lineitem_l_discount_type=number');
```

```
    push (@defparams, 'tab_lineitem_l_tax_type=number');
    push (@defparams, 'tab_lineitem_l_returnflag_type=char(1)');
    push (@defparams, 'tab_lineitem_l_linestatus_type=char(1)');
    push (@defparams, 'tab_lineitem_l_shipdate_type=date');
    push (@defparams, 'tab_lineitem_l_commitdate_type=date');
    push (@defparams, 'tab_lineitem_l_receiptdate_type=date');
    push (@defparams, 'tab_lineitem_l_shipinstruct_type=char(25)');
    push (@defparams, 'tab_lineitem_l_shipmode_type=char(10)');
    push (@defparams, 'tab_lineitem_l_comment_type=varchar(44)');
    push (@defparams, 'tab_lineitem_l_orderkey_pos=13:24');
    push (@defparams, 'tab_lineitem_l_partkey_pos=25:36');
    push (@defparams, 'tab_lineitem_l_suppkey_pos=37:48');
    push (@defparams, 'tab_lineitem_l_linenumber_pos=49:60');
    push (@defparams, 'tab_lineitem_l_quantity_pos=61:72');
    push (@defparams, 'tab_lineitem_l_extendedprice_pos=73:87');
    push (@defparams, 'tab_lineitem_l_discount_pos=88:102');
    push (@defparams, 'tab_lineitem_l_tax_pos=103:117');
    push (@defparams, 'tab_lineitem_l_returnflag_pos=118:118');
    push (@defparams, 'tab_lineitem_l_linestatus_pos=120:120');
    push (@defparams, 'tab_lineitem_l_shipdate_pos=122:131');
    push (@defparams, 'tab_lineitem_l_commitdate_pos=135:144');
    push (@defparams, 'tab_lineitem_l_receiptdate_pos=148:157');
    push (@defparams, 'tab_lineitem_l_shipinstruct_pos=161:185');
    push (@defparams, 'tab_lineitem_l_shipmode_pos=186:195');
    push (@defparams, 'tab_lineitem_l_comment_pos=196:239');
    push (@defparams, 'tab_lineitem_l_shipdate_loadcolx=date
"yyyy-mm-dd"');
    push (@defparams, 'tab_lineitem_l_commitdate_loadcolx=date
"yyyy-mm-dd"');
    push (@defparams, 'tab_lineitem_l_receiptdate_loadcolx=date
"yyyy-mm-dd"');
    push (@defparams, 'tab_orders_parttype=range');
    push (@defparams, 'tab_orders_o_orderkey_type=number');
    push (@defparams, 'tab_orders_o_custkey_type=number');
    push (@defparams, 'tab_orders_o_orderstatus_type=char(1)');
    push (@defparams, 'tab_orders_o_totalprice_type=number');
    push (@defparams, 'tab_orders_o_orderdate_type=date');
    push (@defparams, 'tab_orders_o_orderpriority_type=char(15)');
    push (@defparams, 'tab_orders_o_clerk_type=char(15)');
    push (@defparams, 'tab_orders_o_shippriority_type=number');
    push (@defparams, 'tab_orders_o_comment_type=varchar(79)');
    push (@defparams, 'tab_orders_o_orderkey_pos=13:24');
    push (@defparams, 'tab_orders_o_custkey_pos=25:36');
    push (@defparams, 'tab_orders_o_orderstatus_pos=37:37');
    push (@defparams, 'tab_orders_o_totalprice_pos=39:53');
    push (@defparams, 'tab_orders_o_orderdate_pos=54:63');
    push (@defparams, 'tab_orders_o_orderpriority_pos=67:81');
    push (@defparams, 'tab_orders_o_clerk_pos=82:96');
    push (@defparams, 'tab_orders_o_shippriority_pos=97:108');
    push (@defparams, 'tab_orders_o_comment_pos=109:187');
    push (@defparams, 'tab_orders_o_orderdate_loadcolx=date
"yyyy-mm-dd"');
    push (@defparams, 'tab_partsupp_parttype=range');
    push (@defparams, 'tab_partsupp_ps_partkey_type=number');
    push (@defparams, 'tab_partsupp_ps_suppkey_type=number');
    push (@defparams, 'tab_partsupp_ps_availqty_type=number');
    push (@defparams, 'tab_partsupp_ps_supplycost_type=number');
    push (@defparams,
'tab_partsupp_ps_comment_type=varchar(199)');
    push (@defparams, 'tab_partsupp_ps_partkey_pos=1:12');
    push (@defparams, 'tab_partsupp_ps_suppkey_pos=13:24');
    push (@defparams, 'tab_partsupp_ps_availqty_pos=25:36');
    push (@defparams, 'tab_partsupp_ps_supplycost_pos=37:51');
    push (@defparams, 'tab_partsupp_ps_comment_pos=52:250');
    push (@defparams, 'tab_part_parttype=range');
    push (@defparams, 'tab_part_p_partkey_type=number');
    push (@defparams, 'tab_part_p_name_type=varchar(55)');
    push (@defparams, 'tab_part_p_mfgr_type=char(25)');
    push (@defparams, 'tab_part_p_brand_type=char(10)');
    push (@defparams, 'tab_part_p_type_type=varchar(25)');
    push (@defparams, 'tab_part_p_size_type=number');
    push (@defparams, 'tab_part_p_container_type=char(10)');
    push (@defparams, 'tab_part_p_retailprice_type=number');
    push (@defparams, 'tab_part_p_comment_type=varchar(23)');
    push (@defparams, 'tab_part_p_partkey_pos=1:12');
    push (@defparams, 'tab_part_p_name_pos=13:67');
    push (@defparams, 'tab_part_p_mfgr_pos=68:92');
    push (@defparams, 'tab_part_p_brand_pos=93:102');
    push (@defparams, 'tab_part_p_type_pos=103:127');
    push (@defparams, 'tab_part_p_size_pos=128:139');
    push (@defparams, 'tab_part_p_container_pos=140:149');
    push (@defparams, 'tab_part_p_retailprice_pos=150:164');
    push (@defparams, 'tab_part_p_comment_pos=165:187');
    push (@defparams, 'tab_customer_parttype=range');
    push (@defparams, 'tab_customer_c_custkey_type=number');
    push (@defparams, 'tab_customer_c_name_type=varchar(25)');
    push (@defparams, 'tab_customer_c_address_type=varchar(40)');
    push (@defparams, 'tab_customer_c_nationkey_type=number');
    push (@defparams, 'tab_customer_c_phone_type=char(15)');
    push (@defparams, 'tab_customer_c_acctbal_type=number');
    push (@defparams,
'tab_customer_c_mktsegment_type=char(10)');
    push (@defparams,
'tab_customer_c_comment_type=varchar(117)');
    push (@defparams, 'tab_customer_c_custkey_pos=1:12');
    push (@defparams, 'tab_customer_c_name_pos=13:30');
    push (@defparams, 'tab_customer_c_address_pos=31:70');
    push (@defparams, 'tab_customer_c_nationkey_pos=71:82');
    push (@defparams, 'tab_customer_c_phone_pos=83:97');
    push (@defparams, 'tab_customer_c_acctbal_pos=98:112');
    push (@defparams, 'tab_customer_c_mktsegment_pos=113:122');
    push (@defparams, 'tab_customer_c_comment_pos=123:239');
    push (@defparams, 'tab_supplier_parttype=range');
    push (@defparams, 'tab_supplier_s_suppkey_type=number');
    push (@defparams, 'tab_supplier_s_name_type=char(25)');
    push (@defparams, 'tab_supplier_s_address_type=varchar(40)');
    push (@defparams, 'tab_supplier_s_nationkey_type=number');
    push (@defparams, 'tab_supplier_s_phone_type=char(15)');
    push (@defparams, 'tab_supplier_s_acctbal_type=number');
    push (@defparams,
'tab_supplier_s_comment_type=varchar(101)');
    push (@defparams, 'tab_supplier_s_suppkey_pos=1:12');
    push (@defparams, 'tab_supplier_s_name_pos=13:37');
    push (@defparams, 'tab_supplier_s_address_pos=38:77');
    push (@defparams, 'tab_supplier_s_nationkey_pos=78:89');
    push (@defparams, 'tab_supplier_s_phone_pos=90:104');
    push (@defparams, 'tab_supplier_s_acctbal_pos=105:119');
    push (@defparams, 'tab_supplier_s_comment_pos=120:220');
    push (@defparams, 'tab_nation_parttype=range');
    push (@defparams, 'tab_nation_n_nationkey_type=number');
    push (@defparams, 'tab_nation_n_name_type=char(25)');
    push (@defparams, 'tab_nation_n_regionkey_type=number');
    push (@defparams, 'tab_nation_n_comment_type=varchar(152)');
    push (@defparams, 'tab_nation_n_nationkey_pos=1:12');
    push (@defparams, 'tab_nation_n_name_pos=13:37');
    push (@defparams, 'tab_nation_n_regionkey_pos=38:49');
    push (@defparams, 'tab_nation_n_comment_pos=50:164');
    push (@defparams, 'tab_region_r_regionkey_type=number');
    push (@defparams, 'tab_region_r_name_type=char(25)');
    push (@defparams, 'tab_region_r_comment_type=varchar(152)');
    push (@defparams, 'tab_region_r_regionkey_pos=1:12');
    push (@defparams, 'tab_region_r_name_pos=13:37');
    push (@defparams, 'tab_region_r_comment_pos=38:152');
    push (@defparams,
'load_dbgen_lineitem_output_prefix='.'lineitem');
    push (@defparams, 'load_dbgen_orders_output_prefix='.'orders');
    push (@defparams,
'load_dbgen_customer_output_prefix='.'customer');
    push (@defparams, 'load_dbgen_part_output_prefix='.'part');
```

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


sub wisc
{
   @defparams = ();
   push (@defparams, 'user=wisc');
   push (@defparams, 'passwd=wisc');
   push (@defparams, 'tab_tables=wisc');
   push (@defparams, 'load_field_terminator=whitespace');
   push (@defparams,
'tab_wisc_columns=unique1,unique2,two,four,ten,twenty,hundred,th
ousand,fivethous,tenthous,odd100,even100,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_lo adcolx=integer
external');
   push (@defparams, 'tab_wisc_unique2_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_two_loadcolx=integer external');
   push (@defparams, 'tab_wisc_four_loadcolx=integer external');
   push (@defparams, 'tab_wisc_ten_loadcolx=integer external');
   push (@defparams, 'tab_wisc_twenty_loadcolx=integer external');
   push (@defparams, 'tab_wisc_hundred_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_thousand_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_fivethous_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_tenthous_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_odd100_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_even100_loadcolx=integer
external');
   push (@defparams, 'tab_wisc_stringu1_loadcolx=char(52)');
   push (@defparams, 'tab_wisc_stringu2_loadcolx=char(52)');
   push (@defparams, 'tab_wisc_string4_loadcolx=char(52)');
}

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

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

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

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

```
        $key =$key."!";
        return if ($p =~ /$key/);
    }
                                                    $result = -1;
                                                }
```

Appendix C Acid Scripts

C.1 a_query.sql

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

set serverout on;

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

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

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

exit;
```

C.2 a_query2.sql

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

set serverout on;

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

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

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

exit;
```

C.3 atom.sh

```
#!/bin/ksh
#
# $Header: atom.sh 08-aug-99.13:48:02 mpoess Exp $
#
# atom.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      atom.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#      Performs atomicity tests.
#      Usage: atom.sh [-n iter] [-p prog] [-u usr/pswd] -h
#
#      Options: See usage below
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess     08/08/99 - Creation
#    mpoess     08/08/99 - Creation
#
```

```
. $KIT_DIR/env

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

usage() {

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

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

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

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

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

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

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

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

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

C.4 atranspl.c

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

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

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

  NOTES
    <other useful comments, qualifications, etc.>

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

*/

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

#include "atranspl.h"

/* Declare error handling functions */

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

/* declarations for ORDERS */

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

/* declarations for LINEITEM */

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

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

sb2 l_npricei;

/* other declarations */

int delta = 0;
double rprice;
double cost;

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

int logfile;             /* fdes for logfile for durability (optional)   */
```

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

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

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

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

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

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

time_t curr_time;         /* Current Time                     */

/* OCI handles */

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

/* OCI bind handles */

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

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

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

OCIBind *o_newtprice2_bp = NULL;

OCIBind *o_key2_bp = NULL;

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

char sqlstmt[1024];

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

void usage()
{

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

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

}


void ACIDexit() {

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

}



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

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

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement returned with info.\n");
    if (type)
```

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

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

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

  ACIDexit();

  exit(1);
}

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

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

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

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

  /* argv[1] – Process Number */

  proc_no = atoi(argv[1]);

  /* argv[2] – Number of Streams */

  num_streams = atoi(argv[2]);

  /* argv[3] – Commit? */

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

  /* argv[4] – Delta? */

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

  /* Process  optional parameters */

  argc -= 4;
  argv += 4;

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

```
      slp = atoi(++(argv[0]));
      break;
    default:
      fprintf(stderr, "Unknown argument %s\n", argv[0]);
      usage();
      break;
    }
  }

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

  /* Initialize the cursors etc. */

  (void) ACIDinit();

  /* sleep for some time (triggering) */

  sleep(trig);

  /* start doing the ACID transactions */

  tr_start = gettime();

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

    /* Generate delta if necessary */

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

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

    curr_time = time(NULL);

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

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

    OCIsexec(tpcsvc,curr,errhp,1);

    curr_time = time(NULL);

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

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

    /* Shall we commit? */

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

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

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

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

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

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

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

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

```
  }

  tr_end = gettime();

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

  /*  Disconnect from ORACLE. */

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

  ACIDexit();

  exit(0);
}


void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

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

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

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

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

  /* get username and password */

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

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SER
VER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_
ATTR_USERNAME,
        errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI
_ATTR_PASSWORD,
        errhp);

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESS
ION,errhp);

  /* Enable session parallel dml */

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

  /* Enable session parallel ddl */

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


  /* Make session serializable */

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

  /* Set optimizer_index_cost_adj = 25 */

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

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


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

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

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

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

#endif /* NOLKEY */

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

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

  /* bind variables */


OCIbbname(curr,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),S
QLT_INT);

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

OCIbbname(curr,delta_bp,errhp,":delta",ADR(delta),SIZ(delta),SQL
T_INT);

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

OCIbbname(curr,l_skey_bp,errhp,":l_skey",ADR(l_skey),SIZ(l_ske
y),SQLT_INT);

OCIbbname(curr,l_quan_bp,errhp,":l_quan",ADR(l_quan),SIZ(l_qu
an),SQLT_INT);

OCIbbname(curr,l_newquan_bp,errhp,":l_newquan",ADR(l_newqua
n),
          SIZ(l_newquan),SQLT_INT);

OCIbbname(curr,l_tax_bp,errhp,":l_tax",ADR(l_tax),SIZ(l_tax),SQ
LT_FLT);

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

OCIbbname(curr,l_eprice_bp,errhp,":l_eprice",ADR(l_eprice),SIZ(l
_eprice),
          SQLT_FLT);

OCIbbname(curr,l_neweprice_bp,errhp,":l_neweprice",ADR(l_newe
price),
          SIZ(l_neweprice),SQLT_FLT);


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

OCIbbname(curr,o_newtprice_bp,errhp,":o_newtprice",ADR(o_new
tprice),
          SIZ(o_newtprice), SQLT_FLT);

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

  /* Open & Parse cursor for end values query */
  sprintf((char *) sqlstmt,SQLTXT3);
 OCIStmtPrepare(cure1,errhp,(text *)sqlstmt,strlen((char *)sqlstmt),
            OCI_NTV_SYNTAX,OCI_DEFAULT);

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

  /* bind variables */


OCIbbname(cure1,l_neweprice1_bp,errhp,":l_neweprice",ADR(l_ne
weprice),
          SIZ(l_neweprice),SQLT_FLT);

OCIbbname(cure1,l_newquan1_bp,errhp,":l_newquan",ADR(l_new
quan),
          SIZ(l_newquan),SQLT_INT);

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

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


OCIbbname(cure2,o_newtprice2_bp,errhp,":o_newtprice",ADR(o_n
ewtprice),
          SIZ(o_newtprice),SQLT_FLT);

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

}
```

C.5 atranspl.h

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

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

   DESCRIPTION

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

*/
#ifndef ATRANSPL_H

#define ATRANSPL_H

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

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

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

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

extern int errno;

#ifndef NULL
#define NULL 0
#endif

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

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define UNAME_LEN 64
#define WRITE_BUF_LEN 1024

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

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

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

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

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

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

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

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

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

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

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

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

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

#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree
4)"
#define PDDLTXT "alter session force parallel ddl parallel (degree
4)"
#define OICATXT "alter session set optimizer_index_cost_adj=25"
```

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

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

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

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

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

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

#endif /* ATRANSPL_H */
```

C.6 ckpt.sh

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

. $KIT_DIR/env

sqlplus -s /NOLOG << !

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

C.7 cnt_hist.sql

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

C.8 consist.sh

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

. $KIT_DIR/env

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

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

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

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

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

usage() {

  echo ""
  echo "Usage: $0 [-n iter] [-s number of stream] [-p prog] [-u
usr/pswd] -h"
  echo ""
  echo "-n iter            : number of iterations, default is 100"
```

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

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

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


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

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

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

i=0

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

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

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

i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}${i} | awk '{printf "%d ", $1}'`
echo "The 10 Keys for file $i are: $KEYS"
#for j in `head -10 ${KEY}${i} | awk '{printf "%d ", $1}'`
for j in $KEYS
do
   sqlplus $USER @/dbms/oracle10i/kit/acid/consistency/consist $j
>> $CON1
   echo "------------------------------" >> $CON1
done
   i=`expr $i + 1`
done

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

i=0

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

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

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

wait

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

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

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

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

i=0
while [ $i -lt $STREAM ]
do
KEYS=`head -10 ${KEY}$i | awk '{printf "%d ", $1}'`
#for j in `head -10 ${KEY}$i | awk '{printf "%d ", $1}'`
echo "The keys to check for consistency after the test from file $i
are:"
echo "$KEYS"
for j in $KEYS
do
   sqlplus $USER @/dbms/oracle10i/kit/acid/consistency/consist $j
>> $CON2
   echo "------------------------------" >> $CON2
done
   i=`expr $i + 1`
done
```

C.9 consist.sql

```
Rem
Rem $Header: consist.sql 08-aug-99.16:59:17 mpoess Exp $
Rem
Rem consist.sql
Rem
Rem  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
Rem
```

```
Rem    NAME
Rem      consist.sql - <one-line expansion of the name>
Rem
Rem  DESCRIPTION
Rem       Verifies the consistency of TPC-D database using the
Rem       consistency condition.
Rem
Rem Usage:   sqlplus tpcd/tpcd @consist
Rem
Rem NOTE
Rem REQUIRES PACKAGES prvtotpt and dbmsotpt
rem
Rem    MODIFIED   (MM/DD/YY)
Rem    mpoess     08/08/99 - Creation
Rem    mpoess     08/08/99 - Created
Rem

set verify off
rem set termout on
rem set echo on


REM
REM Get today's date.
REM

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

set serverout on;

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


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

   diff := l_tprice - o_tprice;

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

END;
.
/


spool off
exit
```

C.10 count_tx.sh

```
#!/bin/ksh

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

C.11 d_hist.sql

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


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

drop table history;

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

exit;
```

C.12 end_acid.sh

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

. $KIT_DIR/env

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

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

# get history count

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

# perform the consistency

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

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

cat $ORACLE_HOME/rdbms/log/alert_1g.log >
${DURA_DIR}/alert_1g.log.post_dura 2>&1
```

C.13 iso.sh

```
#!/bin/ksh
#
# $Header: iso.sh 17-aug-99.15:44:51 mpoess Exp $
#
# iso.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#    NAME
#      iso.sh
#
#    DESCRIPTION
#     This script triggers all 6 isolation tests. In addition,
#     it creates more readable formats of the isolation test output.
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      08/17/99 - Creation
#    mpoess      08/17/99 - Creation
#

for iso in iso1 iso2 iso3 iso4 iso5 iso6;do
        echo Running isolation test $iso
        /dbms/oracle10i/kit/acid/isolation/${iso}.sh
        echo Creating nicely formated output of ACID test $iso
        /dbms/oracle10i/kit/acid/isolation/xiso.pl -o
${ACID_OUT}/${iso}
done
```

C.14 iso1.sh

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

. $KIT_DIR/env
```

```
# May need to change the following:
RSH=rsh

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

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

USER=$DATABASE_USER
PROG=atranspl


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

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

usage() {

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

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

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


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

# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 10 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY

echo "Running ACID query 10 seconds AFTER the start of ACID
Transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST}  sqlplus $USER
@$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.15 iso2.sh

```
#!/bin/ksh
#
# $Header: iso2.sh 04-aug-99.09:19:54 mpoess Exp $
#
# iso2.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#   NAME
#     iso2.sh - <one-line expansion of the name>
#
#   DESCRIPTION
#     Usage: iso2.sh [-u user/password] [-n remote_node] -h
#     Options: See usage below
#   NOTES
#     For a cross node isolation test, assume the local node is
#     one of the participating nodes.  The other node can be
#     specified by the -n option.
#     You need to set the environment variable TPCD_KIT_DIR
#
#   MODIFIED   (MM/DD/YY)
#   mpoess     08/04/99 - Creation
#   mpoess     08/04/99 - Creation
#
#
#
================================================
=================+
# May need to change the following:

. $KIT_DIR/env

RSH=rsh

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

```
OUT_DIR=$ACID_OUT

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

randkey 1 0.1 u"$USER" > $KEYFILE

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

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

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 10 seconds before starting ACID query

sleep 10

# start ACID query with the same OKEY

echo "Running ACID query 10 seconds AFTER the start of ACID
transaction" \
>> $TXN2FILE
echo "`date`" >> $TXN2FILE
if [ "$HOST" != "" ]
then
echo "Starting ACID query on node $HOST" >> $TXN2FILE
${RSH} -n ${HOST}  sqlplus "$USER"
@$ACID_DIR/isolation/a_query $OKEY >> $TXN2FILE
else
sqlplus $USER @$ACID_DIR/isolation/a_query $OKEY >>
$TXN2FILE
fi

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.16 iso3.sh

```
#!/bin/ksh
#
# $Header: iso3.sh 04-aug-99.09:20:35 mpoess Exp $
#
# iso3.sh
#
#  Copyright (c) Oracle Corporation 1999. All Rights Reserved.
#
#   NAME
#     iso3.sh - <one-line expansion of the name>
#
#  DESCRIPTION
#     Usage: iso3.sh [-u user/password] [-n remote_node] -h
#     Options: See usage below
#  NOTES
#      For a cross node isolation test, assume the local node is
#      one of the participating nodes.  The other node can be
#      specified by the -n option.
#            We need to make sure the remote node has access to the
#            file system on the local node.  Otherwise, we need to rcp
#            the keyfile to the remote system.
#      You need to set the environment variable TPCD_KIT_DIR
#
#   MODIFIED   (MM/DD/YY)
#   mpoess     08/04/99 - Creation
#   mpoess     08/04/99 - Creation
#

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

```
USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

sleep 1

# start ACID transaction, Sleep for 30 second before COMMIT

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

# let's sleep 10 seconds before starting second ACID transaction

sleep 10

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

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

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.17 iso4.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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

```
  --) break;;
  esac
  shift
done


# generate key files

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

sleep 1

# start ACID transaction, Sleep for 30 second before ROLLBACK

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

# let's sleep 10 seconds before starting second ACID transaction

sleep 10

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

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

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

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.18 iso5.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

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

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

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

sleep 1
```

```
# start ACID transaction, Sleep for 60 second before COMMIT

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

# let's sleep 5 seconds before starting PARTSUPP query

sleep 5

# First generate PS_PARTKEY and PS_SUPPKEY

PSKEY=`randpsup 1`

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

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

wait

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

cat $TXN1FILE $TXN2FILE >> $ISOFILE

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

C.19 iso6.sh

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

. $KIT_DIR/env

# May need to change the following:
RSH=rsh

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

DURA_DIR=$ACID_DIR/dura

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

USER=$DATABASE_USER
PROG=atranspl

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

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

usage() {

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

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

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


# generate key files

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

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

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

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

sleep 1

# start Query 1, use 0 as the delta
```

```
echo "Running Query 21 at `date`" >> $TXN1FILE
sqlplus $USER @$KIT_DIR/acid/isolation/q21 >> $TXN1FILE &

# sleep 2 seconds before starting ACID transaction

sleep 2

# start ACID transaction, COMMIT after one second

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

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

# start Query 1

sleep 2

echo "Running 2nd Query 21 at `date`" >> $TXN3FILE
sqlplus $USER @$KIT_DIR/acid/isolation/q21 >> $TXN3FILE &
# wait for everyone to finish

wait

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

cat $TXN1FILE $TXN2FILE $TXN3FILE >> $ISOFILE

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

C.20 randkey.c

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

/*

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

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

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

#define ORDERCNT 150000.0

/* MK_SPARSE adopted from dss.h */

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

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

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

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

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

/* OCI handles */

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

OCIBind *l_key_bp;
OCIBind *o_key_bp;

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

char sqlstmt[1024];


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


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

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

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement returned with info.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4 *) &errcode,(text
*)msg,
                         2048,OCI_HTYPE_ERROR);
    else
```

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

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

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

  ACIDexit();

  exit(1);
}


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

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

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

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

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

  argc -= 2;
  argv += 2;

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

  ACIDinit();

  /* initialize array for random numbers */

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

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

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

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

    /* Obtain l_key from l_key query */

    OCIsexec(tpcsvc,curi,errhp,1);

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

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

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

  ACIDexit();
  free(res);

}


void usage() {

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

void ACIDinit()
{

  /* run random seed */

  srand48(getpid());

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

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

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

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

 /* get username and password */

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

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SER
VER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,lname,strlen(lname),OCI_
ATTR_USERNAME,
        errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI
_ATTR_PASSWORD,
        errhp);

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


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESS
ION,errhp);

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

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


OCIbbname(curi,l_key_bp,errhp,":l_key",ADR(l_key),SIZ(l_key),S
QLT_INT);

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

C.21 randpsup.c

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

/*

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

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

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

*/

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

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

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

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

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


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

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

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

  /* seed the random number generator */

  srand48(getpid());

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

  PART_SUPP_BRIDGE(ps_skey, pkey, supp);

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

  exit(0);
}
```

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

C.22 sample.sh

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

# $1 durability output file

. $KIT_DIR/env

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



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

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

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

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

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

C.23 sample.sql

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

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

exit;
```

C.24 atrans.sql

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


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

CREATE OR REPLACE PACKAGE d_atrans
IS
PROCEDURE doatrans
(
                l_key                   IN OUT integer,
                o_key                   IN OUT integer,
                delta                   IN OUT integer,
                l_pkey          IN OUT integer,
                l_skey          IN OUT integer,
                l_quan          IN OUT integer,
                l_newquan       IN OUT integer,
                l_tax           IN OUT number,
                l_disc          IN OUT number,
```

```
            l_eprice       IN OUT number,
            l_neweprice    IN OUT number,
            o_tprice    IN OUT number,
            o_newtprice             IN OUT number,
            rprice                  IN OUT number,
            cost                    IN OUT number
);
END;
/

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

   select o_totalprice
            into o_tprice
            from orders
            where o_orderkey = o_key;

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

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

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

   update orders
      set o_totalprice = o_newtprice
      where o_orderkey = o_key;

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

  EXIT;

 EXCEPTION
  WHEN not_serializable THEN
    ROLLBACK;
 END;

 END LOOP;

END doatrans;
END;
/
exit;
```

C.25 run_acid.sh

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

. $KIT_DIR/env

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

usage() {

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

```
  echo "-t trigger  : trigger time between process starts, default is 1
second"
  echo "-h          : print this usage summary"
  exit 1;
}

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

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

# get all the options

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

#collect system info before durability start
cat /var/adm/syslog/syslog.log > ${DURA_DIR}/syslog_pre_dura
2>&1
ps -ef > ${DURA_DIR}/ps.out.pre_dura 2>&1
cat $ORACLE_HOME/rdbms/log/alert_1g.log >
${DURA_DIR}/alert_1g.log.pre_dura 2>&1

echo "Starting ACID run..."

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

# Get history count before the run

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

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

wait
# perform the consistency

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

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

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

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

done

wait

echo "ACID run completed"
```

C.26 prepare4acid.sh

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

sqlplus $DATABASE_USER @d_hist
sqlplus $DATABASE_USER @atrans
```

C.27 q21.sql

```
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 * from (
select
        s_name,
        count(*)  numwait
from
        supplier,
        lineitem l1,
        orders,
        nation
where
        s_suppkey = l1.l_suppkey
        and o_orderkey = l1.l_orderkey
        and o_orderstatus = 'F'
        and l1.l_receiptdate > l1.l_commitdate
        and exists (
                select
                        *
                from
                        lineitem l2
                where
                        l2.l_orderkey = l1.l_orderkey
                        and l2.l_suppkey <> l1.l_suppkey
        )
        and not exists (
                select
                        *
                from
                        lineitem l3
                where
                        l3.l_orderkey = l1.l_orderkey
                        and l3.l_suppkey <> l1.l_suppkey
                        and l3.l_receiptdate > l3.l_commitdate
        )
        and s_nationkey = n_nationkey
        and n_name = 'SAUDI ARABIA'
group by
        s_name
order by
        numwait desc,
        s_name)
where rownum <= 10;

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

exit;
```

Appendix D Query text and Output

D.1 qryqual

Begin Execution at Fri Sep 19 08:04:38 2003

```
-- using default substitutions
-- @(#)1.sql 2.1.6.2
-- TPC-H/TPC-R Pricing Summary Report Query (Q1)
-- Functional Query Definition
-- Approved February 1998


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

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


4 rows processed.
Query Processed in 1.12 seconds.
```

```
-- @(#)2.sql 2.1.6.2
-- TPC-H/TPC-R Minimum Cost Supplier Query (Q2)
-- Functional Query Definition
-- Approved February 1998


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

```
S_ACCTBAL S_NAME N_NAME
P_PARTKEY P_MFGR
S_ADDRESS S_PHONE
S_COMMENT
9938.53 Supplier#000005359 UNITED KINGDOM
185358.00 Manufacturer#4
QKuHYh,vZGiwu2FWEJoLDx04 33-429-790-6131
blithely silent pinto beans are furiously. slyly final deposits acros
9937.84 Supplier#000005969 ROMANIA
108438.00 Manufacturer#1
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa 29-520-692-3537
carefully slow deposits use furiously. slyly ironic platelets above the
ironic
9936.22 Supplier#000005250 UNITED KINGDOM
249.00 Manufacturer#4
B3rqp0xbSEim4Mpy2RH J 33-320-228-2957
blithely special packages are. stealthily express deposits across the
closely final instructi
9923.77 Supplier#000002324 GERMANY
29821.00 Manufacturer#4
y3OD9UywSTOk 17-779-299-1839
quickly express packages breach quiet pinto beans. requ
9871.22 Supplier#000006373 GERMANY
43868.00 Manufacturer#5
J8fcXWsTqM 17-813-485-8637
```

```
never silent deposits integrate furiously blit
9870.78         Supplier#000001286      GERMANY
81285.00        Manufacturer#2
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH   17-516-924-4574
final theodolites cajole slyly special,
<.....deleted>

7912.91         Supplier#000004211      GERMANY
184210.00       Manufacturer#4
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG       17-266-947-7315
final requests integrate slyly above the silent, even
7894.56         Supplier#000007981      GERMANY
85472.00        Manufacturer#4
NSJ96vMROAbeXP                  17-963-404-3760
regular, even theodolites integrate carefully. bold, special theodolites
are slyly fluffily iron
7887.08         Supplier#000009792      GERMANY
164759.00       Manufacturer#3
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz      17-988-938-4296
pending, ironic packages sleep among the carefully ironic accounts.
quickly final accounts
7871.50         Supplier#000007206      RUSSIA
104695.00       Manufacturer#1
3w fNCnrVmvJjE95sgWZzvW         32-432-452-7731
furiously dogged pinto beans cajole. bold, express notornis until the
slyly pending
7852.45         Supplier#000005864      RUSSIA
8363.00         Manufacturer#4
WCNfBPZeSXh3h,c                 32-454-883-3821
blithely regular deposits
7850.66         Supplier#000001518      UNITED KINGDOM
86501.00        Manufacturer#1
ONda3YJiHKJOC                   33-730-383-3892
furiously final accounts wake carefully idle requests. even dolphins
wake acc
7843.52         Supplier#000006683      FRANCE
11680.00        Manufacturer#4
2Z0JGkiv01Y00oCFwUGfviIbhzCdy           16-464-517-8943
carefully bold accounts doub


100 rows processed.
Query Processed in 3.82 seconds.


-- @(#)3.sql            2.1.6.2
-- TPC-H/TPC-R Shipping Priority Query (Q3)
-- Functional Query Definition
-- Approved February 1998


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

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


10 rows processed.
Query Processed in 2.19 seconds.


-- @(#)4.sql            2.1.6.2
-- TPC-H/TPC-R Order Priority Checking Query (Q4)
-- Functional Query Definition
-- Approved February 1998


select
o_orderpriority,
count(*) as order_count
from
orders
where
o_orderdate >= to_date( '1993-07-01',  'YYYY-MM-DD')
and o_orderdate < add_months(to_date( '1993-07-01', 'YYYY-MM-
DD'),3)
and exists (
select
*
from
lineitem
where
l_orderkey = o_orderkey
and l_commitdate < l_receiptdate
)
group by
o_orderpriority
order by
o_orderpriority

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


5 rows processed.
Query Processed in 1.69 seconds.


-- @(#)5.sql            2.1.6.2
-- TPC-H/TPC-R Local Supplier Volume Query (Q5)
-- Functional Query Definition
-- Approved February 1998
```

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

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


5 rows processed.
Query Processed in 4.77 seconds.


-- @(#)6.sql              2.1.6.2
-- TPC-H/TPC-R Forecasting Revenue Change Query (Q6)
-- Functional Query Definition
-- Approved February 1998


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

REVENUE
123141078.23


1 row processed.
Query Processed in 0.15 seconds.


-- @(#)7.sql    2.1.6.2
-- TPC-H/TPC-R Volume Shipping Query (Q7)
-- Functional Query Definition
-- Approved February 1998


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

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


4 rows processed.
Query Processed in 4.67 seconds.


-- @(#)8a.sql           2.1.6.2
-- TPC-H/TPC-R National Market Share Query (Q8)
-- Variant A
-- Approved February 1998


select
o_year,
sum(case when nation='BRAZIL' then volume else 0 end )/
sum(volume)
as mkt_share
from
```

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

O_YEAR	MKT_SHARE
1995.00	0.03
1996.00	0.04

2 rows processed.
Query Processed in 7.95 seconds.

```
-- @(#)9.sql 2.1.6.2
-- TPC-H/TPC-R Product Type Profit Measure Query (Q9)
-- Functional Query Definition
-- Approved February 1998


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

NATION	O_YEAR	SUM_PROFIT
ALGERIA	1998.00	31342867.23
ALGERIA	1997.00	57138193.02
ALGERIA	1996.00	56140140.13
ALGERIA	1995.00	53051469.65
ALGERIA	1994.00	53867582.13
ALGERIA	1993.00	54942718.13
ALGERIA	1992.00	54628034.71
ARGENTINA	1998.00	30211185.71
ARGENTINA	1997.00	50805741.75
ARGENTINA	1996.00	51923746.58

<.....deleted>

UNITED STATES	1998.00	25126238.95
UNITED STATES	1997.00	50077306.42
UNITED STATES	1996.00	48048649.47
UNITED STATES	1995.00	48809032.42
UNITED STATES	1994.00	49296747.18
UNITED STATES	1993.00	48029946.80
UNITED STATES	1992.00	48671944.50
VIETNAM	1998.00	30442736.06
VIETNAM	1997.00	50309179.79
VIETNAM	1996.00	50488161.41
VIETNAM	1995.00	49658284.61
VIETNAM	1994.00	50596057.26
VIETNAM	1993.00	50953919.15
VIETNAM	1992.00	49613838.32

175 rows processed.
Query Processed in 2.79 seconds.

```
-- @(#)10.sql 2.1.6.2
-- TPC-H/TPC-R Returned Item Reporting Query (Q10)
-- Functional Query Definition
-- Approved February 1998


select * from (
select
c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from
customer,
orders,
lineitem,
nation
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and o_orderdate >= to_date ('1993-10-01', 'YYYY-MM-DD')
```

```
and o_orderdate < add_months(  to_date( '1993-10-01',  'YYYY-
MM-DD'),  3)
and l_returnflag = 'R'
and c_nationkey = n_nationkey
group by
c_custkey,
c_name,
c_acctbal,
c_phone,
n_name,
c_address,
c_comment
order by
revenue desc)
where rownum <= 20

C_CUSTKEY           C_NAME                  REVENUE
C_ACCTBAL           N_NAME
C_ADDRESS                       C_PHONE
C_COMMENT
57040.00            Customer#000057040      734235.25
632.87              JAPAN
Eioyzjf4pp                      22-895-641-3466
requests sleep blithely about the furiously i
143347.00           Customer#000143347      721002.69
2557.47             EGYPT
1aReFYv,Kw4                     14-742-935-3718
fluffily bold excuses haggle finally after the u
60838.00            Customer#000060838      679127.31
2454.77             BRAZIL
64EaJ5vMAHWJlBOxJklpNc2RJiWE    12-913-494-9813
furiously even pinto beans integrate under the ruthless foxes; ironic,
even dolphins across the slyl
101998.00           Customer#000101998      637029.57
3790.89             UNITED KINGDOM
01c9CILnNtfOQYmZj               33-593-865-6378
accounts doze blithely! enticing, final deposits sleep blithely special
accounts. slyly express accounts pla
125341.00           Customer#000125341      633508.09
4983.51             GERMANY
S29ODD6bceU8QSuuEJznkNaK        17-582-695-5962
quickly express requests wake quickly blithely
25501.00            Customer#000025501      620269.78
7725.04             ETHIOPIA
 W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ     15-874-808-
6793
quickly special requests sleep evenly among the special deposits.
special deposi
115831.00           Customer#000115831      596423.87
5098.10             FRANCE
rFeBbEEyk dl ne7zV5fDrmiq1oK09wV7pxqCgIc 16-715-386-3788
carefully bold excuses sleep alongside of the thinly idle
84223.00            Customer#000084223      594998.02
528.65              UNITED KINGDOM
nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA      33-442-824-8191
pending, final ideas haggle final requests. unusual, regular
asymptotes affix according to the even foxes.
54289.00            Customer#000054289      585603.39
5583.02             IRAN
vXCxoCsU0Bad5JQI ,oobkZ         20-834-292-4707
express requests sublate blithely regular requests. regular, even ideas
solve.
39922.00            Customer#000039922      584878.11
7321.11             GERMANY
Zgy4s50l2GKN4pLDPBU8m342gIw6R   17-147-757-8036
even pinto beans haggle. slyly bold accounts inte
6226.00             Customer#000006226      576783.76
2230.09             UNITED KINGDOM
8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g,       33-657-701-3391
quickly final requests against the regular instructions wake blithely
final instructions. pa
922.00              Customer#000000922      576767.53
3869.25             GERMANY
Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq         17-945-916-9648
boldly final requests cajole blith
147946.00           Customer#000147946      576455.13
2030.13             ALGERIA
iANyZHjqhyy7Ajah0pTrYyhJ        10-886-956-3143
furiously even accounts are blithely above the furiousl
115640.00           Customer#000115640      569341.19
6436.10             ARGENTINA
Vtgfia9qI 7EpHgecU1X            11-411-543-4901
final instructions are slyly according to the
73606.00            Customer#000073606      568656.86
1785.67             JAPAN
xuR0Tro5yChDfOCrjkd2ol          22-437-653-6966
furiously bold orbits about the furiously busy requests wake across
the furiously quiet theodolites. d
110246.00           Customer#000110246      566842.98
7763.35             VIETNAM
7KzflgX MDOq7sOkI               31-943-426-9837
dolphins sleep blithely among the slyly final
142549.00           Customer#000142549      563537.24
5085.99             INDONESIA
ChqEoK43OysjdHbtKCp6dKqjNyvvi9          19-955-562-2398
regular, unusual dependencies boost slyly; ironic attainments nag
fluffily into the unusual packages?
146149.00           Cust omer#000146149     557254.99
1791.55             ROMANIA
s87fvzFQpU                      29-744-164-6487
silent, unusual requests detect quickly slyly regul
52528.00            Customer#000052528      556397.35
551.79              ARGENTINA
NFztyTOR10UOJ                   11-208-192-3205
unusual requests detect. slyly dogged theodolites use slyly. deposit
23431.00            Customer#000023431      554269.54
3381.86             ROMANIA
HgiV0phqhaIa9aydNoIlb           29-915-458-2654
instructions nag quickly. furiously bold accounts cajol


20 rows processed.
Query Processed in 4.00 seconds.


-- @(#)11.sql           2.1.6.2
-- TPC-H/TPC-R Important Stock Identification Query (Q11)
-- Functional Query Definition
-- Approved February 1998


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

```
supplier,
nation
where
ps_suppkey = s_suppkey
and s_nationkey = n_nationkey
and n_name = 'GERMANY'
)
order by
value desc
```

```
PS_PARTKEY          VALUE
129760.00           17538456.86
166726.00           16503353.92
191287.00           16474801.97
161758.00           16101755.54
34452.00            15983844.72
139035.00           15907078.34
9403.00             15451755.62
154358.00           15212937.88
38823.00            15064802.86
85606.00            15053957.15

<.....deleted>

52338.00            7898638.08
194299.00           7898421.24
105235.00           7897829.94
77207.00            7897752.72
96712.00            7897575.27
10157.00            7897046.25
171154.00           7896814.50
79373.00            7896186.00
113808.00           7893353.88
27901.00            7892952.00
128820.00           7892882.72
25891.00            7890511.20
122819.00           7888881.02
154731.00           7888301.33
101674.00           7879324.60
51968.00            7879102.21
72073.00            7877736.11
5182.00             7874521.73


1048 rows processed.
Query Processed in 4.01 seconds.
```

```
-- @(#)12.sql           2.1.6.2
-- TPC-H/TPC-R Shipping Modes and Order Priority Query (Q12)
-- Functional Query Definition
-- Approved February 1998


select
        l_shipmode,
        sum(case
                when o_orderpriority = '1-URGENT'
                        or o_orderpriority = '2-HIGH'
                        then 1
                else 0
        end) as high_line_count,
        sum(case
                when o_orderpriority <> '1-URGENT'
                        and o_orderpriority <> '2-HIGH'
                        then 1
                else 0
        end) as low_line_count
from
        orders,
        lineitem
where
        o_orderkey = l_orderkey
        and l_shipmode in ('MAIL', 'SHIP')
        and l_commitdate < l_receiptdate
        and l_shipdate < l_commitdate
and l_receiptdate >= to_date( '1994-01-01',  'YYYY-MM-DD')
and l_receiptdate < add_months(to_date ('1994-01-01',  'YYYY-
MM-DD'),  12)
group by
        l_shipmode
order by
        l_shipmode
```

```
L_SHIPMODE HIGH_LINE_COUNT      LOW_LINE_COUNT
MAIL       6202.00              9324.00
SHIP       6200.00              9262.00


2 rows processed.
Query Processed in 1.50 seconds.
```

```
-- @(#)13.sql           2.1.6.2
-- TPC-H/TPC-R Customer Distribution Query (Q13)
-- Functional Query Definition
-- Approved February 1998


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

```
C_COUNT             CUSTDIST
0.00                50004.00
9.00                6641.00
10.00               6566.00
11.00               6058.00
8.00                5949.00
12.00               5553.00
13.00               4989.00
19.00               4748.00
7.00                4707.00
18.00               4625.00
15.00               4552.00
17.00               4530.00
14.00               4484.00
20.00               4461.00
16.00               4323.00
21.00               4217.00
22.00               3730.00
6.00                3334.00
23.00               3129.00
```

```
24.00           2622.00
25.00           2079.00
5.00            1972.00
26.00           1593.00
27.00           1185.00
4.00            1033.00
28.00           869.00
29.00           559.00
3.00            398.00
30.00           373.00
31.00           235.00
2.00            144.00
32.00           128.00
33.00           71.00
34.00           48.00
35.00           33.00
1.00            23.00
36.00           17.00
37.00           7.00
40.00           4.00
38.00           4.00
39.00           2.00
41.00           1.00


42 rows processed.
Query Processed in 1.68 seconds.


-- @(#)14.sql          2.1.6.2
-- TPC-H/TPC-R Promotion Effect Query (Q14)
-- Functional Query Definition
-- Approved February 1998


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

PROMO_REVENUE
16.38


1 row processed.
Query Processed in 1.03 seconds.


-- @(#)15.sql   2.1.6.2
-- TPC-H/TPC-R Top Supplier Query (Q15)
-- Functional Query Definition
-- Approved February 1998


with   revenue
as (select
l_suppkey supplier_no,
sum(l_extendedprice * (1 - l_discount)) total_revenue
from
lineitem
where
l_shipdate >= to_date( '1996-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1996-01-01', 'YYYY-MM-
DD'), 3)
group by
l_suppkey)
select
s_suppkey,
s_name,
s_address,
s_phone,
total_revenue
from
supplier,
revenue
where
s_suppkey = supplier_no
and total_revenue = (
select
max(total_revenue)
from
revenue  )
order by
s_suppkey

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


1 row processed.
Query Processed in 15.72 seconds.


-- @(#)16.sql          2.1.6.2
-- TPC-H/TPC-R Parts/Supplier Relationship Query (Q16)
-- Functional Query Definition
-- Approved February 1998


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

```
p_brand,
p_type,
p_size

P_BRAND    P_TYPE                    P_SIZE
SUPPLIER_CNT
Brand#41   MEDIUM BRUSHED TIN        3.00                28.00
Brand#54   STANDARD BRUSHED COPPER   14.00
27.00
Brand#11   STANDARD BRUSHED TIN      23.00               24.00
Brand#11   STANDARD BURNISHED BRASS  36.00
24.00
Brand#15   MEDIUM ANODIZED NICKEL    3.00
24.00
Brand#15   SMALL ANODIZED BRASS      45.00               24.00
Brand#15   SMALL BURNISHED NICKEL    19.00
24.00
Brand#21   MEDIUM ANODIZED COPPER    3.00
24.00
Brand#22   SMALL BRUSHED NICKEL      3.00                24.00
Brand#22   SMALL BURNISHED BRASS     19.00
24.00
<.....................................................deleted >

Brand#15   LARGE PLATED NICKEL       45.00               3.00
Brand#15   LARGE POLISHED NICKEL     9.00                3.00
Brand#21   PROMO BURNISHED STEEL     45.00               3.00
Brand#22   STANDARD PLATED STEEL     23.00               3.00
Brand#25   LARGE PLATED STEEL        19.00               3.00
Brand#32   STANDARD ANODIZED COPPER  23.00
3.00
Brand#33   SMALL ANODIZED BRASS      9.00                3.00
Brand#35   MEDIUM ANODIZED TIN       19.00               3.00
Brand#51   SMALL PLATED BRASS        23.00               3.00
Brand#52   MEDIUM BRUSHED BRASS      45.00               3.00
Brand#53   MEDIUM BRUSHED TIN        45.00               3.00
Brand#54   ECONOMY POLISHED BRASS    9.00                3.00
Brand#55   PROMO PLATED BRASS        19.00               3.00
Brand#55   STANDARD PLATED TIN       49.00               3.00


18314 rows processed.
Query Processed in 1.68 seconds.


-- @(#)17.sql            2.1.6.2
-- TPC-H/TPC-R Small-Quantity-Order Revenue Query (Q17)
-- Functional Query Definition
-- Approved February 1998


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

AVG_YEARLY
348406.05


1 row processed.
Query Processed in 2.33 seconds.


-- @(#)18.sql            2.1.6.2
-- TPC-H/TPC-R Large Volume Customer Query (Q18)
-- Function Query Definition
-- Approved February 1998


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

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

```
Customer#000146608      146608.00       3342468.00
1994-06-12
499794.58       303.00
Customer#000096103      96103.00        5984582.00
1992-03-16
494398.79       312.00

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

408513.00       305.00
Customer#000017746      17746.00        6882.00
1997-04-09
408446.93       303.00
Customer#000013072      13072.00        1481925.00
1998-03-15
399195.47       301.00
Customer#000082441      82441.00        857959.00
1994-02-07
382579.74       305.00
Customer#000088703      88703.00        2995076.00
1994-01-30
363812.12       302.00


57 rows processed.
Query Processed in 1.26 seconds.


-- @(#)19.sql           2.1.6.2
-- TPC-H/TPC-R Discounted Revenue Query (Q19)
-- Functional Query Definition
-- Approved February 1998


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

REVENUE
3083843.06


1 row processed.
Query Processed in 1.37 seconds.


-- @(#)20.sql           2.1.6.2
-- TPC-H/TPC-R Potential Part Promotion Query (Q20)
-- Function Query Definition
-- Approved February 1998


select
s_name,
s_address
from
supplier,
nation
where
s_suppkey in (
select
ps_suppkey
from
partsupp
where
ps_partkey in (
select
p_partkey
from
part
where
p_name like 'forest%'
)
and ps_availqty > (
select
0.5 * sum(l_quantity)
from
lineitem
where
l_partkey = ps_partkey
and l_suppkey = ps_suppkey
and l_shipdate >= to_date ('1994-01-01', 'YYYY-MM-DD')
and l_shipdate < add_months( to_date ('1994-01-01', 'YYYY-MM-
DD'), 12)
)
)
and s_nationkey = n_nationkey
and n_name = 'CANADA'
order by
s_name

S_NAME                  S_ADDRESS
Supplier#000000020      iybAE,RmTymrZVYaFZva2SH,j
Supplier#000000091
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
Supplier#000000197      YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
Supplier#000000226      83qOdU2EYRdPQAQhEtn GRZEd
Supplier#000000285      Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
Supplier#000000378      FfbhyCxWvcPrO8ltp9
Supplier#000000402      i9Sw4DoyMhzhKXCH9By,AYSgmD
Supplier#000000530      0qwCMwobKY
OcmLyfRXlagA8ukENJv,
Supplier#000000688      D
fw5ocppmZpYBBIPI718hCihLDZ5KhKX

<.....................deleted  >
```

```
Supplier#000009796      z,y4Idmr15DOvPUqYG
Supplier#000009799       4wNjXGa4OKWl
Supplier#000009811      E3iuyq7UnZxU7oPZIe2Gu6
Supplier#000009812
APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
Supplier#000009862      rJzweWeN58
Supplier#000009868      ROjGgx5gvtkmnUUoeyy7v
Supplier#000009869
ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
Supplier#000009899      7XdpAHrzr1t,UQFZE
Supplier#000009974      7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT


204 rows processed.
Query Processed in 1.70 seconds.
```

```
-- @(#)21.sql           2.1.6.2
-- TPC-H/TPC-R Suppliers Who Kept Orders Waiting Query (Q21)
-- Functional Query Definition
-- Approved February 1998


select * from (
select
s_name,
count(*)  numwait
from
supplier,
lineitem l1,
orders,
nation
where
s_suppkey = l1.l_suppkey
and o_orderkey = l1.l_orderkey
and o_orderstatus = 'F'
and l1.l_receiptdate > l1.l_commitdate
and exists (
select
*
from
lineitem l2
where
l2.l_orderkey = l1.l_orderkey
and l2.l_suppkey <> l1.l_suppkey
)
and not exists (
select
*
from
lineitem l3
where
l3.l_orderkey = l1.l_orderkey
and l3.l_suppkey <> l1.l_suppkey
and l3.l_receiptdate > l3.l_commitdate
)
and s_nationkey = n_nationkey
and n_name = 'SAUDI ARABIA'
group by
s_name
order by
numwait desc,
s_name)
where rownum <= 100
```

```
S_NAME                NUMWAIT
Supplier#000002829      20.00
Supplier#000005808      18.00
Supplier#000000262      17.00
Supplier#000000496      17.00
Supplier#000002160      17.00
Supplier#000002301      17.00

<..... deleted >

Supplier#000000673      12.00
Supplier#000000762      12.00
Supplier#000000811      12.00
Supplier#000000821      12.00
Supplier#000001337      12.00
Supplier#000001916      12.00
Supplier#000001925      12.00
Supplier#000002039      12.00
Supplier#000002357      12.00
Supplier#000002483      12.00

100 rows processed.
Query Processed in 17.67 seconds.
```

```
-- @(#)22.sql   2.1.4.2
-- TPC-H/TPC-R Global Sales Opportunity Query (Q22)
-- Functional Query Definition
-- Approved February 1998


select
cntrycode,
count(*) as numcust,
sum(c_acctbal) as totacctbal
from
(
select
substr(c_phone, 1, 2) as cntrycode,
c_acctbal
from
customer
where
substr(c_phone,1, 2) in
('13', '31', '23', '29', '30', '18', '17')
and c_acctbal > (
select
avg(c_acctbal)
from
customer
where
c_acctbal > 0.00
and substr(c_phone, 1, 2) in
('13', '31', '23', '29', '30', '18', '17')
)
and not exists (
select
*
from
orders
where
o_custkey = c_custkey
)
) custsale
group by
cntrycode
order by
cntrycode
```

```
CNTRYCODE NUMCUST          TOTACCTBAL
13        888.00           6737713.99
17        861.00           6460573.72
18        964.00           7236687.40
23        892.00           6701457.95
```

29	948.00	7158866.63
30	909.00	6808436.13
31	922.00	6806670.18

7 rows processed.
Query Processed in 1.15 seconds.

Ended Executing this Stream at Fri Sep 19 08:06:02 2003

Stream Started at 1063983878.14
Stream Ended at 1063983962.41
Stream Processed in 84.26 seconds

SQL statements processed: 22

Appendix E Seed and Input Parameters

E.1 seed

0917033726

E.2 stream00

14 1994-12-01
2 50 NICKEL MIDDLE EAST
9 aquamarine
20 plum 1994-01-01 INDIA
6 1996-01-01 0.08 24
17 Brand#25 SM PKG
18 315
8 BRAZIL AMERICA STANDARD ANODIZED COPPER
21 INDIA
13 pending requests
3 FURNITURE 1995-03-26
22 34 30 20 11 32 13 14
16 Brand#22 ECONOMY BRUSHED 1 10 17 35 29 41 14 40
4 1997-01-01
11 ALGERIA0.0000000333
15 1997-09-01
1 119
10 1993-03-01
19 Brand#14 Brand#22 Brand#33 3 14 25
5 ASIA 1996-01-01
7 CHINA BRAZIL
12 TRUCK RAIL 1994-01-01

E.3 stream01

21 ALGERIA
3 MACHINERY 1995-03-11
18 312
5 EUROPE 1996-01-01
11 JORDAN 0.0000000333
7 IRAN ROMANIA
6 1996-01-01 0.05 25
20 burlywood1993-01-01 RUSSIA
17 Brand#22 SM DRUM
12 RAIL TRUCK 1995-01-01
16 Brand#12 SMALL BURNISHED 26 2 13 32 25 24 39 8
15 1995-06-01
13 pending requests
10 1993-12-01
2 38 TIN AMERICA
8 ROMANIA EUROPE PROMO POLISHED COPPER
14 1995-04-01
19 Brand#11 Brand#15 Brand#32 8 15 21
9 violet
22 32 25 31 13 17 29 12
1 66
4 1994-10-01

E.4 stream02

6 1996-01-01 0.02 24
17 Brand#33 LG BAG
14 1995-07-01
16 Brand#43 LARGE PLATED 13 12 17 1 22 6 16 25
19 Brand#23 Brand#43 Brand#31 4 16 28
10 1994-09-01
9 spring
2 26 STEEL MIDDLE EAST
15 1993-02-01
8 IRAQ MIDDLE EAST PROMO BURNISHED COPPER
5 MIDDLE EAST 1996-01-01
22 22 23 12 17 27 21 34
12 AIR RAIL 1995-01-01
7 BRAZIL IRAQ
13 pending requests
18 314
1 74
4 1997-05-01
20 metallic 1996-01-01 JAPAN
3 BUILDING 1995-03-28
11 ARGENTINA 0.0000000333
21 PERU

E.5 stream03

8 CANADA AMERICA PROMO ANODIZED TIN
5 AFRICA 1997-01-01
4 1995-01-01
6 1997-01-01 0.08 24
17 Brand#35 LG PKG
7 ROMANIA CANADA
1 82
18 315
22 19 18 11 13 27 26 15
14 1995-10-01
9 seashell
10 1993-06-01
15 1995-09-01
11 KENYA 0.0000000333
20 violet 1994-01-01 BRAZIL
2 14 BRASS ASIA
21 INDONESIA
19 Brand#25 Brand#21 Brand#25 9 17 25
13 pending accounts
16 Brand#23 PROMO BRUSHED 4 49 20 5 18 26 12 36
12 REG AIR TRUCK 1995-01-01
3 HOUSEHOLD 1995-03-13

E.6 stream04

5 AMERICA 1997-01-01
21 ARGENTINA
14 1996-01-01
19 Brand#22 Brand#14 Brand#25 4 18
21
15 1993-06-01
17 Brand#32 LG DRUM
12 FOB TRUCK 1995-01-01
6 1997-01-01 0.05 25
4 1997-08-01
9 rose
8 SAUDI ARABIA MIDDLE EAST ECONOMY
POLISHED TIN
16 Brand#13 MEDIUM ANODIZED 20
21 5 11 15 9
31 46
11 BRAZIL 0.0000000333
2 2 NICKEL MIDDLE EAST
10 1994-03-01
18 313
1 90
13 pending accounts
7 IRAQ SAUDI ARABIA
22 27 21 15 18 12
24 11
3 BUILDING 1995-03-30
20 grey 1993-01-01 PERU

E.7 stream05

21 ROMANIA
15 1996-01-01
4 1995-05-01
6 1997-01-01 0.03 24
7 CANADA JAPAN
16 Brand#43 ECONOMY PLATED 24
17 34 14 19 22
25 46
19 Brand#34 Brand#42 Brand#24 9 19
28
18 314
14 1996-05-01
22 23 15 25 11 28
12 17
11 MOROCCO 0.0000000333
13 unusual accounts
3 HOUSEHOLD 1995-03-15
1 98
2 39 TIN ASIA
5 ASIA 1997-01-01
8 JAPAN ASIA ECONOMY BURNISHED TIN
20 saddle 1996-01-01 FRANCE
12 MAIL TRUCK 1996-01-01
17 Brand#34 MED BAG
10 1995-01-01
9 pink

E.8 stream06

10 1993-10-01
3 AUTOMOBILE 1995-03-01
15 1993-09-01
13 unusual accounts
6 1997-01-01 0.08 24
8 EGYPT MIDDLE EAST LARGE BRUSHED
TIN
9 orange
7 SAUDI ARABIA EGYPT
4 1993-02-01
11 CANADA 0.0000000333
22 31 24 22 21 34
16 15
18 312
12 TRUCK MAIL 1993-01-01
1 106
5 EUROPE 1997-01-01
16 Brand#33 STANDARD POLISHED 31
11 29 12 8 25
17 15
2 27 COPPER AFRICA
14 1996-08-01
19 Brand#32 Brand#35 Brand#13 5 20
24
20 cream 1995-01-01 VIETNAM
17 Brand#31 MED PKG
21 IRAQ

E.9 stream07

18 314
8 VIETNAM ASIA LARGE PLATED TIN
20 orange 1993-01-01 IRAQ
21 CANADA
2 15 BRASS ASIA
4 1995-09-01
22 29 15 18 16 26
14 33
17 Brand#33 MED DRUM
1 114
11 MOZAMBIQUE 0.0000000333
9 mint
19 Brand#34 Brand#13 Brand#12 10 10
20
3 HOUSEHOLD 1995-03-17
13 unusual deposits
5 AFRICA 1993-01-01
7 JAPAN VIETNAM
10 1994-07-01
16 Brand#13 LARGE ANODIZED 24 16 7
9 23 38 10 45
6 1993-01-01 0.06 25
14 1996-11-01
15 1996-04-01
12 RAIL MAIL 1996-01-01

E.10 stream08

19 Brand#41 Brand#51 Brand#11 5 11
28
1 61
15 1994-01-01
17 Brand#35 JUMBO BAG
5 AMERICA 1993-01-01
8 JORDAN MIDDLE EAST LARGE ANODIZED
NICKEL
9 linen
12 AIR MAIL 1997-01-01
14 1997-02-01

```
7       EGYPT   JORDAN
4       1993-06-01
3       AUTOMOBILE      1995-03-03
20      beige   1997-01-01      ALGERIA
16      Brand#43        PROMO BURNISHED         12      4
        43      6       50      46      30
        16
6       1993-01-01      0.03    25
22      16      34      14      22      26
        13      23
10      1993-04-01
13      unusual deposits
2       3       NICKEL  AFRICA
21      SAUDI ARABIA
18      315
11      EGYPT   0.0000000333
```

Appendix F Benchmark Scripts

F.1 dbtables.sql

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

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



SELECT * FROM REGION;

SELECT COUNT(*) FROM NATION;

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



SELECT COUNT(*) FROM ORDERS;

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



SELECT COUNT(*) FROM PART;

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



SELECT COUNT(*) FROM PARTSUPP;

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

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

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

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

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



SELECT COUNT(*) FROM SUPPLIER;

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



DROP TABLE MINMAX;

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

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

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

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

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

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

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

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

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

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

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

SELECT * FROM MINMAX;
```

```
spool off
exit;
```

F.2 gen_seed.sh

```
#!/bin/ksh

SEED_FILE=$1

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

F.3 gtime.c

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

/*

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

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

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

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

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

   NOTES
     <other useful comments, qualifications, etc.>

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

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

# include <sys/time.h>

main ()
{

  struct timeval tv;

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

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

}

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

F.4 qexecpl.c

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

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

/*

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

   DESCRIPTION
     SQL Execution Engine, Oracle v8,  OCI version

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

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

*/

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

#include "qexecpl.h"

/* Function Prototypes */

extern double gettime();

/* function prototypes from gen.c */

int get_statement();

/* Declare error handling functions */

void sql_error();

/* Other prototypes */

int define_output_variables();
void process_select_list();
void usage();
void SQLinit();
```

```
void SQLexec();
void SQLexit();
void *memalloc();
void print_header();
void print_rows();
int OFEN();
void remove_newline();

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

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

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

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

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

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

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

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

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

char stmt[SQL_LEN];       /* The SQL statement or comment line.
*/
char qn[3];               /* Number of the query being executed  */
char qnp[3];              /* Number of the previous query executed  */
char cmnt[5000];          /* Buffer to save the comment.         */
#ifdef LINUX
FILE *qtemp;      /* fd for query template               */
FILE *logfile;    /* log and report files                */
FILE *rep;
#else
FILE *qtemp = stdin;      /* fd for query template               */
FILE *logfile = stdout;   /* log and report files                */
FILE *rep = stdout;
#endif
void *defbuf;             /* Buffer pointer for ODEFIN           */
int deflen = 0;           /* Size of data type for ODEFIN        */
int deftype = 1;          /* Oracle type number for ODEFIN       */

int pfmem = PFMEMSIZE;    /* Memory to prefetch rows             */

time_t tim;               /* To get wall clock time              */

/* OCI handles */

OCIEnv *tpcenv = NULL;
OCIServer *tpcsrv = NULL;
OCIError *errhp = NULL;
OCISvcCtx *tpcsvc = NULL;
OCISession *tpcusr = NULL;
OCIStmt *curq = NULL;
OCIStmt *cur_dml = NULL;
OCIStmt *cur_ddl = NULL;
OCIParam *tpcpar = NULL;

sword status = OCI_SUCCESS;  /* OCI return value */

/* usage: prints the usage of the program */

void usage() {

  fprintf(stderr,"\nUsage: qexec username/password [q<path name
for query template file>] \n");
  fprintf(stderr,"               [l<path name for log>] [r<path name for
reports>] \n\n");
  fprintf(stderr,"Options:\n");
  fprintf(stderr,"q<path for query>         : full path name for the query
template file.\n");
  fprintf(stderr,"                          (default is stdin)\n");
  fprintf(stderr,"l<path name for log>      : full path name for log
files\n");
  fprintf(stderr,"                          (default is stdout)\n");
  fprintf(stderr,"r<path name for reports>  : full path name for
reports\n");
  fprintf(stderr,"                          (default is stdout)\n");
  exit(-1);
}



/* type: 0 if environment handle is passed, 1 if error handle is
passwd */

void sql_error(errhp,status,type)
     OCIError *errhp;
     sword status;
     sword type;
{
  char msg[2048];
  ub4 errcode;
  ub4 msglen;
  int i,j;

  switch(status) {
  case OCI_SUCCESS_WITH_INFO:
    fprintf(stderr, "Error: Statement returned with in fo.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_ERROR:
    fprintf(stderr, "Error: OCI call error.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  case OCI_INVALID_HANDLE:
    fprintf(stderr, "Error: Invalid Handle.\n");
    if (type)
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
                                        2048,OCI_HTYPE_ERROR);
    else
      (void) OCIErrorGet(errhp,1,NULL,(sb4*)&errcode,(text*)msg,
```

```
                                        2048,OCI_HTYPE_ENV);
    fprintf(stderr,"%s\n",msg);
    break;
  }

  /* Rollback just in case */

  (void) OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);

  fprintf(stderr, "Exiting Oracle...\n");
  fflush(stderr);

  SQLexit();

  exit(1);
}

#ifdef LINUX
int main(argc,argv)
#else
void main(argc,argv)
#endif
     int argc;
     char *argv[];
{

  int i,pos,pos2;
  int retcode;      /* Return code for get_statement */
#ifdef LINUX
  logfile=fopen("/dev/stdout","w");
  qtemp=fopen("/dev/stdin","rw");
  rep=fopen("/dev/stdout","w");
#endif
  /* Initialize some variables */

  if ((argc > 5) || (argc < 2)) {
    usage();
  }

  /* argv[1] -- User and Password for Database */

  strcpy(logname, argv[1]);

  /* Process  optional parameters */

  argc -= 1;
  argv += 1;

  while(--argc) {
    ++argv;
    switch(argv[0][0]) {
    case 'q':
      if ((qtemp = fopen(++(argv[0]),"r")) == NULL) {
              fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
              fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
              exit(-1);
      }
      break;
    case 'r':
      if ((rep = fopen(++(argv[0]),"a")) == NULL) {
              fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
              fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
              exit(-1);
      }
      break;
    case 'l':
      if ((logfile = fopen(++(argv[0]),"a")) == NULL) {
              fprintf(stderr,"Unable to open file '%s'\n", argv[0]);
              fprintf(stderr,"%s: %s\n", argv[0], strerror(errno));
              exit(-1);
      }
      break;
    default:
      fprintf(stderr,"Invalid Option: %c\n", argv[0][0]);
      usage();
      break;
    }
  }

  /* Do some initialization and establish connection with the database
*/

  SQLinit();

  /* May want to add some triggering mechanism here */

  time(&tim);
  fprintf(logfile, "Begin Execution at %s\n\n", ctime(&tim));
  fprintf(rep, "Begin Executing this Stream at %s\n\n", ctime(&tim));
  /* Get the next statement and start processing it */

  while ((retcode = get_statement()) > 0) {

    switch (retcode) {

      /* If this is a comment, skips it */
    case COMMENT:
      /*if (end_flag) {
              end_flag = 0;  */    /* reset query end flag */
              /* save the comment so that we can print it out later on  */
              /* strcpy(cmnt, stmt);
              break;
      } */
      if (stmt[3]== '@') {
        pos=4;
        strcpy(qnp,qn);
        while (stmt[pos] != ')') {
          pos++;
        }
        pos2=0;
        pos++;
        while (stmt[pos] != '.') {
          /*printf ("qn %d %c \n",pos2,stmt[pos]);*/
          qn[pos2]=stmt[pos];
          pos2++;
          pos++;
        }
        qn[pos2] = 0;
        /* printf("found a new query: %s\n",qn); */
      }
      /* save the comment so that we can print it out later on  */
      strcat(cmnt, stmt);
      break;

      /* if this is a set_row_fetch command */
    case SET_FETCHROW:
      fprintf(logfile,"Setting the number of rows to fetch to: %ld\n\n",
                    num_to_fetch);
      break;

      /* if this is a SQL statement */
    case SQL_STMT:

      /* Executes the query */
      SQLexec();

      stmt_cnt++;
      qry_cnt++;
      fflush(rep);
      fflush(logfile);
```

```
    /*
    fprintf(logfile,"\nStatement Started at %.2f\n", s_tr_start);
    fprintf(logfile,"Statement Ended at %.2f\n", s_tr_end);

    fprintf(logfile,"Statement Processed in %.2f seconds.\n",
                (s_tr_end - s_tr_start));
    fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
            qn,(s_tr_end - s_tr_start)s_tr_start,s_tr_end);
    fflush(rep);
    fflush(logfile);*/
    break;

    /* Should never reach here */
   default:
    fprintf(stderr, "Invalid statement type!!\n");
    SQLexit();
    break;
   }
  }

  /* Get Timing for the last query */

  tr_end = gettime();

  fprintf(logfile,"Query Processed in %.2f seconds.\n\n",(tr_end -
s_tr_start));

  /* print comments for this query that we have saved */

  /* fprintf(logfile, "%s\n", cmnt); */

  /* fprintf(rep, "Query %s : Execution time %.2f\n", qn,(tr_end -
s_tr_start));*/
  fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
            qn,(tr_end - s_tr_start),s_tr_start,tr_end);

  time(&tim);
  fprintf(logfile,"\nEnded Executing this Stream at %s\n",
ctime(&tim));
  fprintf(logfile,"\nStream Started at %.2f\n", tr_start);
  fprintf(logfile,"Stream Ended at %.2f\n", tr_end);
  fprintf(logfile,"Stream Processed in %.2f seconds\n\n",(tr_end -
tr_start));

  fprintf(rep,"\nEnded Executing this Stream at %s\n", ctime(&tim));
  fprintf(rep,"\nStream Started at %.2f\n", tr_start);
  fprintf(rep,"Stream Ended at %.2f\n", tr_end);
  fprintf(rep,"Stream Processed in %.2f seconds\n\n",
            (tr_end - tr_start));

  fprintf(logfile, "\nSQL statements processed: %d\n", stmt_cnt);
  /*fprintf(logfile, "Queries processed: %d\n", qry_cnt);*/

  fflush(rep);
  fflush(logfile);

  /* Close the query template file */

  fclose(qtemp);

  /*  Disconnect from ORACLE. */

  SQLexit();
  exit(0);
}


/* SQLinit(): Perform initialization tasks.                    */
/*        Logs on to Oracle, opens some files and open a cursor for
*/
/*        later use.                                           */

void SQLinit() {

  int i;

  /* preallocate MAX_PREALLOC members of the dlist array
*/
  /* initializes others to NULL so that we can determine who to free
later */

  for (i=0; i<MAX_SEL_LIST; i++) {
    if (i < MAX_PREALLOC) {
      dlist[i] = (dltype *) memalloc (sizeof(dltype));
      dlist[i]->defhdl = NULL;
/*      OCIhalloc(curq,&(dlist[i]->defhdl),OCI_HTYPE_DEFINE);
*/
    }
    else
      dlist[i] = NULL;
  }

  /* Connect to ORACLE.  Program will call sql_error()        */
  /* if an error occurs in connecting to the default database. */

  (void) OCI Initialize(OCI_DEFAULT,(dvoid *)0,0,0,0);

  if((status=OCIEnvInit((OCIEnv
**)&tpcenv,OCI_DEFAULT,0,(dvoid **)0)) !=
     OCI_SUCCESS)
    sql_error(tpcenv, status, 0);

  OCIhalloc(tpcenv,&errhp,OCI_HTYPE_ERROR);
  OCIhalloc(tpcenv,&curq,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cur_dml,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&cur_ddl,OCI_HTYPE_STMT);
  OCIhalloc(tpcenv,&tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhalloc(tpcenv,&tpcsrv,OCI_HTYPE_SERVER);
  OCIhalloc(tpcenv,&tpcusr,OCI_HTYPE_SESSION);

  /* get username and password */

  passwd = strchr(logname, '/');
  *passwd = '\0';
  passwd++;

  if ((status = OCIServerAttach(tpcsrv,errhp,(text
*)0,0,OCI_DEFAULT)) != OCI_SUCCESS)
    sql_error(errhp,status,1);


OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcsrv,0,OCI_ATTR_SER
VER,errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,logname,strlen(logname),O
CI_ATTR_USERNAME,
            errhp);

OCIaset(tpcusr,OCI_HTYPE_SESSION,passwd,strlen(passwd),OCI
_ATTR_PASSWORD,
            errhp);

  if ((status = OCISessionBegin(tpcsvc, errhp, tpcusr,
OCI_CRED_RDBMS,
                                        OCI_DEFAULT)) !=
OCI_SUCCESS)
    sql_error(errhp,status,1);
```

```
OCIaset(tpcsvc,OCI_HTYPE_SVCCTX,tpcusr,0,OCI_ATTR_SESS
ION,errhp);

/*
  if ((status=OCILogon((OCIEnv *)tpcenv,(OCIError
*)errhp,(OCISvcCtx *)tpcsvc,
                       (text *)logname, strlen(logname), (text
*)passwd,
                       strlen(passwd), (text *) 0, 0)) !=
OCI_SUCCESS)
    sql_error(errhp, status, 1);
*/
  printf("\nConnected to ORACLE as user: %s\n\n", logname);

}



/* SQLexec()  Executes the SQL statement.                    */
/*           Parse the SQL statement.                        */
/*           If DDL or DML statements, execute right away.   */
/*           Else describe and define select list outputs,   */
/*               execute and fetch results.                  */

void SQLexec()
{
  int i;
  ub2 stmttyp = OCI_STMT_SELECT;     /* default is a SELECT
statement */

  /* Clause 5.3.6.2: QI(i,s) is the time between the first character */
  /*               of this query text is submitted and the first   */
  /*               character of the next query text is submitted.  */

  if (qry_cnt) {
    time(&tim);
     s_tr_end = gettime();
     fprintf(logfile,"Query Processed in %.2f seconds.\n\n",
      (s_tr_end - s_tr_start));

     /* print comments for this query that we have saved */

     /* fprintf(logfile, "%s\n", cmnt); */

     /*fprintf(rep, "Query %s : Execution time %.2f\n", qnp,(s_tr_end
- s_tr_start));*/
     fprintf(rep, "Query %s: Execution Time: %.2f started %.2f ended
%.2f\n",
                qnp,(s_tr_end - s_tr_start),s_tr_start,s_tr_end);

     /* Let's fflush stuff so that we can see what's going on */

     fflush(logfile);
     fflush(rep);
  }
  else
     tr_start = gettime();

  s_tr_start = gettime();

  /* prepare the statement */

  if ((status = OCIStmtPrepare(curq, errhp, (text*) stmt, (ub4)
strlen(stmt),
                                          OCI_NTV_SYNTAX,
OCI_DEFAULT)) != OCI_SUCCESS)
     sql_error(errhp,status,1);

  /* Prints the query text and comment to the logfile */

  fprintf(logfile, "\n%s\n", cmnt);
  cmnt[0]=0;
  fprintf(logfile, "\n%s\n", stmt);

  /* if this is a DDL or DML statement, execute it right away */
  /* only worries about SELECT statements right now, cannot   */
  /* execute a stored PL/SQL procedure in thie version        */


OCIaget(curq,OCI_HTYPE_STMT,&stmttyp,NULL,OCI_ATTR_S
TMT_TYPE,errhp);

  if (stmttyp != OCI_STMT_SELECT) {
    OCIsexec(tpcsvc,curq,errhp,1);
    return;
  }

  /* otherwise, this is a select statement */
  /* Describe and define output variables  */

  /* first let's execute it to get the select-list definition */

  OCIaset(curq, OCI_HTYPE_STMT, &pfmem, 0,
OCI_ATTR_PREFETCH_MEMORY, errhp);

  OCIsexec(tpcsvc,curq,errhp,0);

  num_sel_list = define_output_variables();

  /* Executes the query and fetches the rows */

  (void) process_select_list(num_sel_list);

  /* Need to get the number of rows fetched first   */
  /* since the following statments will screw it up */


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_
ROW_COUNT,errhp);

  /* To control memory usage, let's free up the extra dlist entries */
  /* that we have allocated.                                        */

  i=MAX_PREALLOC;
  while(dlist[i] != NULL) {
    free(dlist[i]);
    dlist[i++] = NULL;
  }

  /* reset set_fetchrows */

  num_to_fetch = -1;

}


void SQLexit() {

  int i;

  OCILogoff(tpcsvc,errhp);
  OCIhfree(tpcenv,OCI_HTYPE_STMT);
  OCIhfree(tpcsvc,OCI_HTYPE_SVCCTX);
  OCIhfree(tpcsrv,OCI_HTYPE_SERVER);
  OCIhfree(tpcusr,OCI_HTYPE_SESSION);

  /* free all memory */
```

```
  for (i=0; i<MAX_SEL_LIST; i++) {
   if (dlist[i] != NULL) {
     free(dlist[i]);
    }
  }

 /* Flush all output */

 fflush(rep);
 fflush(logfile);

}


/* define_output_variables(): Describe and define select-list items
for */
/*                        a query statement.                    */
/*                        Returns the number of select-list items   */
/*                        for this query.                       */

int define_output_variables()
{

  int i;
  int retflag = 0;

  for (i=0; i<MAX_SEL_LIST; i++) {

    slist[i].buflen = MAX_COLNAME_SIZE;

   if (OCIParamGet(curq, OCI_HTYPE_STMT, errhp, (dvoid **)
&tpcpar,
                                        POS(i)) != OCI_SUCCESS)
     break;


    /* dsize and nullok fields of dlist not used */

   OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbsize),
                NULL, OCI_ATTR_DATA_SIZE, errhp);
   OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].dbtype),
                NULL, OCI_ATTR_DATA_TYPE, errhp);
   OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].buf),
                &(slist[i].buflen), OCI_ATTR_NAME, errhp);
   OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].precision),
                NULL, OCI_ATTR_PRECISION, errhp);
   OCIaget(tpcpar, OCI_DTYPE_PARAM, &(slist[i].scale),
                NULL, OCI_ATTR_SCALE, errhp);

   /* For formatting purpose, remove trailing blanks in select-list
name. */

/*
   if (slist[i].buflen < MAX_COLNAME_SIZE)
     (slist[i].buf)[slist[i].buflen] = '\0';
*/
   /* Well, we need to allocate for entries for dlist */

   if (i >= MAX_PREALLOC) {
     dlist[i] = (dltype *) memalloc(sizeof(dltype));
     dlist[i]->defhdl = NULL;
    }

   /* Let's check the sizes and types for this select list item */

    switch (slist[i].dbtype) {

   case OCI_TYPECODE_NUMBER:

     /* The odescr will not give a good estimate to the scale if */
     /* no scale was given in the Oracle table definition.       */

#ifdef HAVE_SCALE
     if (slist[i].scale != 0) {
               defbuf = (double *) dlist[i]->fbuf;
               deflen = FLT;
               deftype = OCI_TYPECODE_DOUBLE;
               slist[i].dbtype = OCI_TYPECODE_DOUBLE;
      } else {
               defbuf = (int *) dlist[i]->ibuf;
               deflen = INT;
               deftype = OCI_TYPECODE_INTEGER;
               slist[i].dbtype = OCI_TYPECODE_INTEGER;
      }
#else
      defbuf = (double *) dlist[i]->fbuf;
      deflen = FLT;
      deftype = OCI_TYPECODE_FLOAT;
      slist[i].dbtype = OCI_TYPECODE_FLOAT;
#endif /* HAVE_SCALE */

      break;

    default:

      /* default is character string */

      defbuf = (char **) dlist[i]->sbuf;
      deflen = MAX_STR_LEN;
      deftype = SQLT_STR;
/*     deftype = OCI_TYPECODE_CHAR; */
      break;
    }

   /* Define the column */

   if ((status=OCIDefineByPos(curq,&(dlist[i]-
>defhdl),errhp,POS(i),
                                        defbuf,deflen,deftype,NULL,
                                        dlist[i]-
>rlen,NULL,OCI_DEFAULT))!=OCI_SUCCESS)
     sql_error(errhp,status,1);
  }
  return i;
}


/* process_select_list(): Fetch rows from a query.               */

void process_select_list(num)
    int num;        /* number of select list items */
{

  int i,j;
  int ntf;
  int num_so_far;
  sword stats = OCI_SUCCESS;

  /* Print the headers for the query execution result */

  print_header(num);

  /* See if we need to limit the rows to fetch */

  ntf = (num_to_fetch >= 0) ? num_to_fetch : MAX_ARRAY;

  /* Fetch the rows and print them out */

  if ((ntf > MAX_ARRAY) || (num_to_fetch == -1)) {
```

```
   stats = OCIStmtFetch(curq, errhp, MAX_ARRAY,
OCI_FETCH_NEXT, OCI_DEFAULT);


OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_
ROW_COUNT,errhp);

   print_rows(num,rows_ret);

   /* To avoid 1022 from OFEN */
   /* More rows to fetch...   */

   if (stats != OCI_NO_DATA) {
    if (num_to_fetch == -1) {
            while ((stats =
OCIStmtFetch(curq,errhp,MAX_ARRAY,OCI_FETCH_NEXT,
                                                OCI_DEFAULT))
== OCI_SUCCESS) {

OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                              OCI_ATTR_ROW_COUNT,errhp);
              print_rows(num,(num_so_far-rows_ret));
              rows_ret = num_so_far;
            }
            /* Print the final rows */
            OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL
,
                              OCI_ATTR_ROW_COUNT,errhp);
            print_rows(num,(num_so_far-rows_ret));
            rows_ret = num_so_far;
     } else {
            ntf -= MAX_ARRAY;

            while ((stats = OCIStmtFetch(curq,errhp,

((ntf>MAX_ARRAY) ? MAX_ARRAY:ntf),

OCI_FETCH_NEXT, OCI_DEFAULT)) ==
                  OCI_SUCCESS) {
              ntf -= MAX_ARRAY;

OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL,
                               OCI_ATTR_ROW_COUNT,errhp);
              print_rows(num,(num_so_far-rows_ret));
              rows_ret = num_so_far;
              if (ntf <= 0) break;
            }
            OCIaget(curq,OCI_HTYPE_STMT,&num_so_far,NULL
,
                              OCI_ATTR_ROW_COUNT,errhp);
            print_rows(num,(num_so_far-rows_ret));
            rows_ret = num_so_far;
     }
    }
  } else {
   OCIStmtFetch(curq, errhp, ntf, OCI_FETCH_NEXT,
OCI_DEFAULT);

OCIaget(curq,OCI_HTYPE_STMT,&rows_ret,NULL,OCI_ATTR_
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 = ' ';
}
```

F.5 qexecpl.h

```
/*
 * $Header: qexecpl.h 13-nov-2001.17:52:35 mpoess Exp $
 */

/* Copyright (c) 1999, 2001, Oracle Corporation.  All rights
reserved.  */

/* NOTE:  See 'header_template.doc' in the 'doc' dve under the
'forms'
     directory for the header file template that includes instructions.
*/

/*
   NAME
     qexecpl.h

   DESCRIPTION
     SQL statement execution front-end header file.

   PUBLIC FUNCTION(S)
     <list of external functions declared/defined - with one-line
descriptions>

   PRIVATE FUNCT ION(S)
     <list of static functions defined in .c file - with one-line
descriptions>

   EXAMPLES
```

```
  NOTES
    <other useful comments, qualifications, etc.>

  MODIFIED   (MM/DD/YY)
  mpoess     11/13/01 - change DOP to 84 for DML and DDL
  mpoess     02/22/01 - add linux changes
  mpoess     08/05/99 - make compile
  mpoess     07/15/99 - Creation
  mpoess     07/15/99 - Creation

*/

/*
# ifndef S_ORACLE
#  include <s.h>
# endif
*/
#ifndef QSTREAMPL_H

#define QSTREAMPL_H

#include <stdio.h>
#include <string.h>
#include <sys/param.h>
#include <sys/types.h>
#include <time.h>
#include <errno.h>
#include <math.h>

#include <oratypes.h>

#include <oratypes.h>

#ifndef OCIDFN
#include <ocidfn.h>
#endif /* OCIDFN */

#ifndef OCI_ORACLE
#include <oci.h>
#endif /* OCI_ORACLE */
/*
#ifdef __STDC__
#include <ociapr.h>
#else
#include <ocikpr.h>
#endif *//* __STDC__ */

/* some basic definitions */

#define UNAME_LEN 64
#define MAX_FILE_PATH_LEN 128

#ifndef TRUE
#define TRUE 1
#endif /* TRUE */

#ifndef FALSE
#define FALSE 1
#endif /* FALSE */
#ifndef LINUX
#define MAX(x,y) ((x >= y) ? x : y)
#define MIN(x,y) ((x <= y) ? x : y)
#endif
/* defines and typedefs for parsing */

#define CRT_TBL  1
#define INS_STMT 3
#define SEL_STMT 4
#define UPD_STMT 5
#define DRP_VIEW 7
#define DRP_TBL  8
#define DEL_STMT 9
#define CRT_VIEW 10

/* defines and typedefs for query description */

#define MAX_COLNAME_SIZE 32   /* Maximum length of
Column name  */
#define MAX_SEL_LIST 16       /* Maximum items on a select list
*/

#define END_OF_LIST 1007      /* Error code when we reach the
end of the */
                              /* select list.                  */

/* types for describe */

#define CHAR_TYPE 1
#define NUM_TYPE  2
#define INT_TYPE  3
#define FLT_TYPE  4
#define STR_TYPE  5
#define DATE_TYPE 12

#define NUMWIDTH 16           /* Width of the numeric fields    */

#define POS(i) (i+1)          /* The position is 1...n instead  */
#define IND(i) (i-1)          /* of 0..n-1 as in an array.      */

typedef struct des
{
  ub2 dbsize;
  ub4 buflen;
/*  sb2 dsize;  */
  sb4 scale;
/*  sb2 nullok; */
  OCITypeCode dbtype;
/*  text buf[MAX_COLNAME_SIZE]; */
  text *buf;
  ub1 precision;
} sltype;


/* defines and typedefs for query select list definition */

#define MAX_ARRAY 50          /* Maximum array size for array
fetch */
#define PFMEMSIZE 65536       /* Memory size of prefetch buffer
*/

#define MAX_STR_LEN 256       /* Maximum size for string
variables          */
#define MAX_PREALLOC 8        /* Maximum number of
preallocated select list */
                              /* definitions.                  */

#define INT sizeof(long)
#define STR sizeof(char)
#define FLT sizeof(double)

#define FLTP (double *)
#define INTP (long *)
#define STRP (char **)

typedef struct def
{
  long ibuf[MAX_ARRAY];
  double fbuf[MAX_ARRAY];
  char sbuf[MAX_ARRAY][MAX_STR_LEN];
  ub2 rlen[MAX_ARRAY];        /* return length     */
```

```
  OCIDefine *defhdl;
} dltype;

extern int errno;

#define SQL_LEN 2048

#ifndef NULL
#define NULL 0
#endif

#ifndef NULLP
# define NULLP (void *)NULL
#endif /* NULLP */

#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#ifndef ub1
#define ub1 unsigned char
#endif

#define NA            -1      /* ANSI SQL NULL */
#define VER7           2
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction
not serializable */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))
#define SID(sid) ((sid == -1) ? 0 : sid)

/* For get_statement */

#define END_OF_FILE -1
#define COMMENT 1
#define SQL_STMT   2
#define SET_FETCHROW 3


#define OCIhalloc(envh,hndl,htyp) \
   if((status=OCIHandleAlloc((dvoid *)envh,(dvoid
**)hndl,htyp,0,(dvoid **)0))!=OCI_SUCCESS) \
      sql_error(envh,st atus,0); \
   else \
     DISCARD 0

#define OCIhfree(hndl,htyp) \
   if((status=OCIHandleFree((dvoid *)hndl,htyp)) ==
OCI_SUCCESS) \
     fprintf(stderr, "Error freeing handle of type %d\n", htyp)

#define OCIaget(hndl,htyp,attp,size,atyp,errh) \
   if((status=OCIAttrGet((dvoid *)hndl,htyp,(dvoid *)attp,(dvoid
*)size,atyp,errh)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
     DISCARD 0

#define OCIaset(hndl,htyp,attp,size,atyp,errh) \
   if((status=OCIAttrSet((dvoid *)hndl,htyp,(dvoid
*)attp,size,atyp,errh)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
     DISCARD 0

#define OCIsexec(svch,stmh,errh,iter) \

if((status=OCIStmtExecute(svch,stmh,errh,iter,0,NULL,NULL,OCI
_DEFAULT)) != OCI_SUCCESS) \
      sql_error(errh,status,1); \
   else \
     DISCARD 0


#define ISOTXT "alter session set isolation_level = serializable"
#define PDMLTXT "alter session force parallel dml parallel (degree
84)"
#define PDDLTXT "alter session force parallel ddl parallel (degree
84)"

#endif /* QSTREAMPL_H */
```

F.6 runTPCHall

```
#!/bin/ksh
. $KIT_DIR/env

ECHO=echo

sqlplus=$ORACLE_HOME/bin/sqlplus
GTIME=${KIT_DIR}/utils/gtime

RUN_ID_FILE=${KIT_DIR}/audit/r_id

if [ ! -f $RUN_ID_FILE ]
then
  echo "0" > $RUN_ID_FILE
fi

RUN_ID=`cat $RUN_ID_FILE`
RUN_ID=`expr $RUN_ID + 1`
echo $RUN_ID > $RUN_ID_FILE

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
  mkdir $OUT_DIR
fi

SCRIPT_LOG_FILE=${OUT_DIR}/main.out
RDB_TABLES=${OUT_DIR}/rdbtablest
FIRST_TEN=${OUT_DIR}/firstten
LD1DBCRE=${OUT_DIR}/Ld1dbcre
LD2SCTSO=${OUT_DIR}/Ld2sctso
LD3DAPOP=${OUT_DIR}/Ld3dapop
LD4IXCRE=${OUT_DIR}/Ld4ixcre
LD5ANLYZ=${OUT_DIR}/Ld5anlyz
DAT_FILE=/dbms/oracle10i/kit/audit/3TB.dat

echo Start TPC-H Benchmark SEQUENCE NUMBER: $RUN_ID >
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "Starting a new Oracle log file:
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log" >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

mv $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log.preAudi
t.$RUN_ID
```

```
touch $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log

echo "Start: load database `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -x -o ${DAT_FILE} -p dbcre > $LD1DBCRE
bumpx.pl -s -x -o ${DAT_FILE} -p sctso > $LD2SCTSO
STIME=`$GTIME`
echo "Start: timed load portion `date`" >> $SCRIPT_LOG_FILE
bumpx.pl -s -x -o ${DAT_FILE} -p dapop > $LD3DAPOP
bumpx.pl -s -x -o ${DAT_FILE} -p ixcre > $LD4IXCRE
bumpx.pl -s -x -o ${DAT_FILE} -p anlyz > $LD5ANLYZ
$KIT_DIR/audit/tshut
$KIT_DIR/audit/tstart
$KIT_DIR/audit/ckpnt.sh
echo "End: timed load portion `date`" >> $SCRIPT_LOG_FILE

$KIT_DIR/audit/gen_seed.sh $KIT_DIR/audit/seed
echo Generated seed: `cat $KIT_DIR/audit/seed` >>
$SCRIPT_LOG_FILE

echo "Start: dbtables.sql and count.sql" >> $SCRIPT_LOG_FILE
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/dbtables >
${RDB_TABLES} 2>&1
$sqlplus ${DATABASE_USER} @$KIT_DIR/audit/firstten >
${FIRST_TEN} 2>&1
echo "End: dbtables.sql and count.sql `date`" >>
$SCRIPT_LOG_FILE

$KIT_DIR/audit/tshut >>  $SCRIPT_LOG_FILE
$KIT_DIR/audit/tstart >>  $SCRIPT_LOG_FILE
$KIT_DIR/audit/ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 1 ${RUN_ID}

$KIT_DIR/audit/tshut >>  $SCRIPT_LOG_FILE
$KIT_DIR/audit/tstart >>  $SCRIPT_LOG_FILE
$KIT_DIR/audit/ckpnt.sh
runTPCHpt ${SCALE_FACTOR} 2 ${RUN_ID}

sleep 600
# call the auditor: don't tshut >> $SCRIPT_LOG_FILE

cp $ORACLE_HOME/rdbms/log/alert_${ORACLE_SID}.log
$OUT_DIR

echo "End TPC-H Benchmark SEQUENCE NUMBE R: $RUN_ID
`date`" >> $SCRIPT_LOG_FILE
```

F.7 runTPCHpt

```
#!/bin/ksh
. $KIT_DIR/env
#set -x
#ECHO=/bin/echo
SCRIPT_DIR=${KIT_DIR}/scripts
UPD_DIR=${KIT_DIR}/update
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of the query
template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen
QEXEC=${SRC_DIR}

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

UPD_SQL=${UPD_DIR}/sql
UPD_SPT=${UPD_DIR}/scripts
UPD_SRC=${UPD_DIR}/source
UPD_DAT=${UPD_DIR}/data
TPCD_BIN=${KIT_DIR}/audit/bin

GTIME=${SRC_DIR}/gtime
SEED_FILE=${KIT_DIR}/audit/seed

DF=/dev/null
HID=1
INTERVAL=60
COUNT=1200

# The defaults

QPROG=${QEXEC}/qexec

usage () {

echo " "
echo "Usage: $0 [-p <program for query stream>] [-u1 <program for
UF1>]"
echo "         [-u2 <program for UF2>] [-o] [-s] [-h] [-u
<user/password>]"
echo "         <scale factor> <run_number>"
echo ""
echo "scale factor     : The scale factor of the run."
echo "update ||ism     : The parallelism to use for the UFs."
echo ""
echo "-p <program>     : Program for Query Stream."
echo "                   Default is $QPROG."
echo "-u1 <program>    : Program for UF1."
echo "                   Default is $U1PROG."
echo "-u2 <program>    : Program for UF2."
echo "                   Default is $U2PROG."
echo "-o               : Collect Oracle statistics."
echo "-s               : Collect System statistics."
echo "-u <user/passwd> : User/Password. Default is tpch/tpch."
echo "-h               : Displays this message."
}
set -- `getopt "p:u1:u2:osu:h" "$@"` || usage

while :
do
  case "$1" in
  -u1) shift; U1PROG=$1;;
  -u2) shift; U2PROG=$1;;
  -p) shift; QPROG=$1;;
# not needed ?    -o) OSTAT=1;;
# not needed ?   -s) SST AT=1;;
  -h) usage; exit 0;;
  --) shift; break;;
  esac
  shift;
done


if [ "$#" -ne "3" ]
then
  usage
  exit 1
fi

SF=$1
PARA=$2
RUN_ID=$3

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
        mkdir $OUT_DIR
fi
```

```
TPCD_LOG=${OUT_DIR}
TPCD_RPT=${OUT_DIR}
OUT=${OUT_DIR}

let UF_SET="($PARA-1)*($NUM_STREAMS+1)+1"
START_SET=1
let STOP_SET=$NUM_STREAMS
let START_SET_UPDATE="($PARA-
1)*($NUM_STREAMS+1)+2"
let
STOP_SET_UPDATE="$START_SET_UPDATE+$NUM_STREA
MS-1"

TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s0
TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s0inter
QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s0
QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.0
SCRIPT_LOG_FILE=${TPCD_LOG}/m${PARA}timing
UF1_LOG=${TPCD_LOG}/m${PARA}s0rf1
UF2_LOG=${TPCD_LOG}/m${PARA}s0rf2
STREAM_COUNT_LOG=${TPCD_LOG}/m${PARA}tstrcnt

echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID}
`date`" > $SCRIPT_LOG_FILE
echo "TPC-H Test - RUN:${PARA} SEQUENCE:${RUN_ID}
`date`" > $TPCD_RPT_FILE
echo "Generates query template file with seed: `cat $SEED_FILE`
for stream 0" >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

${QGEN} -c -r `cat $SEED_FILE` -p 0 -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}

START=`$GTIME`
echo "Start Power Test - RUN:${PARA} SEQUENCE:${RUN_ID}
Execution Starts $START, `date`" >>  $SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

# Execute UF1

SDATE=`date`
UF1_START=`$GTIME`
echo "Start UF1  $UF1_START, `date`" >> $SCRIPT_LOG_FILE

${ECHO} ${UPD_SPT}/runuf1.sh ${UF_SET} >> $UF1_LOG
2>&1
# Execute Query Stream

UF1_END=`${GTIME}`
E1DATE=`date`

UF1_TIME=`echo $UF1_END - $UF1_START | bc`
echo UF1: Execution Time: $UF1_TIME >> ${TPCD_RPT_FILE}
echo Start Time: $UF1_START, $SDATE >> ${TPCD_RPT_FILE}
echo End Time: $UF1_END, $E1DATE >> ${TPCD_RPT_FILE}
echo "" >> ${TPCD_RPT_FILE}

echo "End UF1 $UF1_END, ${E1DATE}" >>
$SCRIPT_LOG_FILE
echo UF1: Execution Time: $UF1_TIME >> $SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

echo "Start Query Part `$GTIME`, `date` " >>
$SCRIPT_LOG_FILE

${QPROG} ${DATABASE_USER} q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE}  > $DF  2>&1

# Execute UF2

UF2_START=`${GTIME}`
E2DATE=`date`

echo "End Query Part `$GTIME`, ${E2DATE}" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE

echo "Start UF2  $UF2_START, `date`" >> $SCRIPT_LOG_FILE
${ECHO} ${UPD_SPT}/runuf2.sh ${UF_SET} >> $UF2_LOG
2>&1
UF2_END=`${GTIME}`
END=`${GTIME}`
EDATE=`date`

UF2_TIME=`echo $UF2_END - $UF2_START | bc`
echo UF2: Execution Time: $UF2_TIME >> ${TPCD_RPT_FILE}
echo Start Time: $UF2_START, $E2DATE >>
${TPCD_RPT_FILE}
echo End Time: $UF2_END, $EDATE >> ${TPCD_RPT_FILE}


echo "End UF2 $UF2_END, $EDATE" >>  $SCRIPT_LOG_FILE
echo UF2: Execution Time: $UF2_TIME >> $SCRIPT_LOG_FILE
echo >>  $SCRIPT_LOG_FILE

echo "End TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID}, $END, $EDATE" >>
$SCRIPT_LOG_FILE
MEA_INT=`echo $END - $ST ART | bc`
echo "Elapsed Time for TPC-H Power Test - RUN:${PARA}
SEQUENCE:${RUN_ID} is $MEA_INT" >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE

${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}

i=$START_SET
PSEED=`cat $SEED_FILE`

while [ $i -le $STOP_SET ]; do
 TPCD_LOG_FILE=${TPCD_LOG}/mt${RUN_ID}_${i}.log
 TPCD_RPT_FILE=${TPCD_RPT}/mt${RUN_ID}_${i}.rpt
          QUERY_PARAMETER=${TPCD_LOG}/qp${PARA}.
${i}
 QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}

 PSEED=`expr $PSEED + 1`
 ${QGEN}  -c -r${PSEED} -p ${i} -s ${SF} -l
$QUERY_PARAMETER > ${QRY_FILE}

 i=`expr $i + 1`
done

TH_START_D=`date`
TH_START_T=`${GTIME}`
echo >> $SCRIPT_LOG_FILE

rm -f /tmp/th_pipe1
mknod /tmp/th_pipe1 p
rm -f /tmp/th_pipe2
mknod /tmp/th_pipe2 p
i=$START_SET

echo "Start Throughput Test - RUN:${PARA}
SEQUENCE:${RUN_ID} $TH_START_T, $TH_START_D"  >>
$SCRIPT_LOG_FILE

# starts a script to count the streams during the throughtput run
(scnt.sh $PARA $RUN_ID > $STREAM_COUNT_LOG &)
```

```
while [ $i -le $STOP_SET ]; do
  M_SDATE=`date`
  M_STIME=`${GTIME}`
            TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
            TPCD_RPT_FILE=${TPCD_RPT}/m${PARA}s${i}int
er
  echo "Start Query Stream $i $M_STIME, ${M_SDATE}" >>
$SCRIPT_LOG_FILE
            QRY_FILE=${TPCD_RPT}/qtemp.${PARA}s${i}
  ${QPROG} ${DATABASE_USER} q${QRY_FILE}
l${TPCD_LOG_FILE} r${TPCD_RPT_FILE} | grep -v "Connected
to ORACLE" >> $SCRIPT_LOG_FILE &
  i=`expr $i + 1`
done

(${KIT_DIR}/audit/runTPCHus $RUN_ID
$START_SET_UPDATE $STOP_SET_UPDATE ${SF} $PARA
>> $SCRIPT_LOG_FILE 2>&1 &)

wait
THQ_END_T=`$GTIME`
THQ_END_D=`date`
echo End all Query Streams $THQ_END_T, $THQ_END_D >>
$SCRIPT_LOG_FILE
print > /tmp/th_pipe1
read < /tmp/th_pipe2

TH_END_D=`date`
TH_END_T=`$GTIME`
echo End Update Stream ${TH_END_T}, ${TH_END_D} >>
$SCRIPT_LOG_FILE
echo >> $SCRIPT_LOG_FILE
echo "End Throughput Test ${TH_END_T}, ${TH_END_D}" >>
$SCRIPT_LOG_FILE
echo Execution Time Throughput Test: `echo ${TH_END_T} -
${TH_START_T} | bc` >> $SCRIPT_LOG_FILE

i=$START_SET
while [ $i -le $STOP_SET ]; do
            TPCD_LOG_FILE=${TPCD_LOG}/m${PARA}s${i}
  ${KIT_DIR}/audit/abridge.pl ${TPCD_LOG_FILE}
            i=`expr $i + 1`
done
PIDS=`ps -fu oracle | grep scnt.sh | grep -v grep | awk '{print $2}'`
kill -9 $PIDS
#calculate the metric
#analyze_streams.pl -f p -n $RUN_ID >
${TPCD_RPT}/tpch_metric.${RUN_ID}.${HID}.rpt
```

F.8 runTPCHus

```
#!/bin/ksh
. $KIT_DIR/env

SCRIPT_DIR=${KIT_DIR}/scripts
SQL_DIR=${KIT_DIR}/sql
UPD_DIR=${KIT_DIR}/update
UPD_SPT=${UPD_DIR}/scripts
SRC_DIR=${KIT_DIR}/utils
QRY_DIR=${KIT_DIR}/queries # this is the location of the query
template file
QGEN_DIR=${KIT_DIR}/dbgen
QGEN=${QGEN_DIR}/qgen

DSS_QUERY=${KIT_DIR}/queries
export DSS_QUERY

RUN_ID=$1
START_SET_UPDATE=$2
STOP_SET_UPDATE=$3
SF=$4
PARA=$5

OUT_DIR=${KIT_DIR}/audit/tests/${RUN_ID}
if [ ! -d $OUT_DIR ]
then
            mkdir $OUT_DIR
fi


TPCD_RPT=$OUT_DIR
SCRIPT_LOG_FILE=${OUT_DIR}/m${PARA}timing
OUT=$OUT_DIR

GTIME=${SRC_DIR}/gtime
HID=1

START=`$GTIME`
echo "Start Update Stream $START, `date`" >>
$SCRIPT_LOG_FILE
echo "" >> $SCRIPT_LOG_FILE


#waiting for all the query streams to finish first
read < /tmp/th_pipe1

i=$START_SET_UPDATE
j=1
while [ $i -le $STOP_SET_UPDATE ]; do

  # Execute UF1

  UF1_LOG=${OUT_DIR}/m${PARA}s${j}rf1
  UF2_LOG=${OUT_DIR}/m${PARA}s${j}rf2
  RPT_FILE=${OUT_DIR}/m${PARA}s${j}inter

  SDATE=`date`
  UF1_START=`$GTIME`
            echo "Start UF1 -${j} at ${UF1_START}, ${SDATE}"
>> ${RPT_FILE}

  ${UPD_SPT}/runuf1.sh ${i} >> ${UF1_LOG} 2>&1
  UF1_END=`${GTIME}`
            EDATE=`date`
  echo "End UF1 -${j} at ${UF1_END}, ${EDATE}" >>
${RPT_FILE}
            echo UF1-${j} Execution Time: `echo ${UF1_END} -
${UF1_START} | bc` >> ${RPT_FILE}

  # Execute UF2

  SDATE=`date`
  UF2_START=`${GTIME}`
  echo "Start UF2 -${j} ${UF2_START}, ${SDATE}" >>
${RPT_FILE}

  ${UPD_SPT}/runuf2.sh ${i} >> ${UF2_LOG} 2>&1
  UF2_END=`${GTIME}`
  EDATE=`date`
  echo "End UF2 -${j} at $UF2_END, ${EDATE}" >>
${RPT_FILE}
            echo UF2-${j} Execution Time: `echo ${UF2_END} -
${UF2_START} | bc` >> ${RPT_FILE}

  i=`expr $i + 1`
            j=`expr $j + 1`
done
```

```
print > /tmp/th_pip e2
```

F.9 runuf1.sh

```
#!/bin/ksh
#
# $Header: runuf1.sh 25-oct-2001.15:56:04 mpoess Exp $
#
# runuf1.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#    NAME
#      runuf1.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf1.sh -l [<path name for reports>] -u [<uid/passwd>]
#              -p [<program>] <run_id> <scale factor> <pair number>
#              <parallelism>
# USAGE
#    To execute UF1.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess      10/25/01 - change default directory for update sets
#    mpoess      10/17/01 - add support for external tables
#    mpoess      08/15/99 - Creation
#    mpoess      08/15/99 - Creation
#
#
. $KIT_DIR/env
O=${ORACLE_HOME}
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
LOG_DIR=${UPDATE_DIR}/log
GTIME=${UTILS_DIR}/gtime
SF=${SCALE_FACTOR}
PAR_HINT=${UPDATE_1_DOP}

LOGPATH=.
PASSWD=${DATABASE_USER}

if [ $# -lt 1 ];
then
   echo runuf1.sh setnum
   exit 1
fi
SETNUM=$1
i=1
PID=""

# perform the update function 1

START=`$GTIME`

# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as '/dbms/oracle10i/kit/update/update_sets';

drop table temp_l_et;
create table temp_l_et(
    l_orderkey          number ,
    l_partkey           number ,
    l_suppkey           number ,
    l_linenumber        number ,
    l_quantity          number ,
    l_extendedprice     number ,
    l_discount          number ,
    l_tax               number ,
    l_returnflag        char(1) ,
    l_linestatus        char(1) ,
    l_shipdate          date ,
    l_commitdate        date ,
    l_receiptdate       date ,
    l_shipinstruct      char(25) ,
    l_shipmode          char(10) ,
    l_comment           varchar(44)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
              records delimited by newline
              badfile 'l_et.${SETNUM}.bad'
              logfile 'l_et.${SETNUM}.log'
              fields terminated by '|'
              missing field values are null
        )
        location (
        'lineitem.tbl.u${SETNUM}'
))
reject limit unlimited;

drop table temp_o_et;
create table temp_o_et(
    o_orderkey          number ,
    o_custkey           number ,
    o_orderstatus       char(1) ,
    o_totalprice        number ,
    o_orderdate         date ,
    o_orderpriority     char(15) ,
    o_clerk             char(15) ,
    o_shippriority      number ,
    o_comment           varchar(79)
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
              records delimited by newline
              badfile 'o_et.${SETNUM}.bad'
              logfile 'o_et.${SETNUM}.log'
              fields terminated by '|'
              missing field values are null
        )
        location (
        'orders.tbl.u${SETNUM}'
))
reject limit unlimited;

alter table temp_l_et parallel  ${PAR_HINT};
alter table temp_o_et parallel  ${PAR_HINT};

alter session force parallel dml parallel (degree ${PAR_HINT});
```

```
alter session set isolation_level = serializable;
alter session set  optimizer_index_cost_adj=10;


insert into orders
select
    o_orderdate         ,
    o_orderkey          ,
    o_custkey           ,
    o_orderpriority     ,
    o_shippriority      ,
    o_clerk             ,
    o_orderstatus       ,
    o_totalprice        ,
    o_comment
from temp_o_et;


insert into  lineitem
select
    l_shipdate          ,
    l_orderkey          ,
    l_discount          ,
    l_extendedprice     ,
    l_suppkey           ,
    l_quantity          ,
    l_returnflag        ,
    l_partkey           ,
    l_linestatus        ,
    l_tax               ,
    l_commitdate        ,
    l_receiptdate       ,
    l_shipmode          ,
    l_linenumber        ,
    l_shipinstruct      ,
    l_comment
from temp_l_et;

commit;

drop table temp_l_et;
drop table temp_o_et;

exit;
!

END=`$GTIME`

# Done

echo ""
echo "Update Function 1 Set $SETNUM done!"
echo "Elapsed Tim e is `echo $END - $START | bc`"
echo ""
```

F.10 runuf2.sh

```
#!/bin/ksh
#
# $Header: runuf2.sh 25-oct-2001.15:56:05 mpoess Exp $
#
# runuf2.sh
#
# Copyright (c) 1999, 2001, Oracle Corporation.  All rights reserved.
#
#    NAME
#      runuf2.sh - <one-line expansion of the name>
#
# DESCRIPTION
#    runuf2.sh [-u <uid/passwd to login>] [-p <program>] <run_id>
#              <scale factor> <pair number> <parallelism>
# USAGE
#   To execute UF2.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    mpoess     10/25/01 - change default directory for update sets
#    mpoess     10/17/01 - add support for external tables
#    mpoess     08/15/99 - Creation
#    mpoess     08/15/99 - Creation
#
. $KIT_DIR/env
UPDATE_DIR=${KIT_DIR}/update
SCRIPT_DIR=${UPDATE_DIR}/scripts
UTILS_DIR=${KIT_DIR}/utils
GTIME=${UTILS_DIR}/gtime
LOG_DIR=${UPDATE_DIR}/log
PAR_HINT=${UPDATE_2_DOP}
SF=${SCALE_FACTOR}
PASSWD=${DATABASE_USER}

if [ $# -lt 1 ]
then
   usage
   exit 1
fi

SETNUM=$1

i=1
PID=""

START=`$GTIME`
# first create the temp tables

sqlplus /NOLOG << !

connect $PASSWD;
set timing  on
set serveroutput on
set echo on

drop directory data_dir;
create directory data_dir as '/dbms/oracle10i/kit/update/update_sets';

drop table temp_okey_et;
drop table temp_okey;

create table temp_okey_et(
    t_orderkey          number
)
organization external (
type ORACLE_LOADER
default directory data_dir
access parameters
(
              records delimited by newline
               badfile 'okey.${SETNUM}.bad'
               logfile 'okey.${SETNUM}.log'
              fields terminated by '|'
             missing field values are null
       )
       location (
       'delete.${SETNUM}'))
reject limit unlimited;
```

```
alter table temp_okey_et parallel 4;

create table temp_okey parallel  4 nologging as select * from
temp_okey_et;

create unique index i_temp_okey on temp_okey (t_orderkey)
parallel 4  nologging compute statistics;

analyze table temp_okey estimate statistics sample 2 percent;

alter session  force parallel dml parallel ${PAR_HINT};
alter session set isolation_level=serializable;
alter session set optimizer_index_cost_adj=10;


delete from (select /*+ use_nl(o) */ o.rowid from orders o,
temp_okey t where o.o_orderkey = t.t_orderkey order by 1);

delete from (select /*+ use_nl(l) */ l.rowid from lineitem
l,temp_okey t where l.l_orderkey = t.t_orderkey order by 1);

commit;

drop table temp_okey;
drop table temp_okey_et;
exit;
!

END=`$GTIME`

# Done

echo ""
echo "Update Function 2 Set $SETNUM done!"
echo "Elapsed Time is `echo $END - $START | bc`"
echo ""
```

F.11 audit_stream.sh

F.12 tstart

```
#!/bin/ksh


mpsched -P RR sqlplus /NOLOG << !
connect / as sysdba
startup pfile=/oracle/dbs/3TB_run.ora
exit
!
/Lvm/set_queue;
exit
```

F.13 tshut

```
#!/bin/ksh

sqlplus /NOLOG << !
connect / as sysdba
shutdown
exit
!
```

F.14 set_queue

```
#!/sbin/sh

set -x

for c in c2  c3 c4 c7 c11 c10 \
        c12 c15 c16 c18 c21 c22 \
        c26 c27 c29 c30 c32 c34 \
        c38 c39 c41 c42 c45 c47 \
        c50 c51 c52 c53 c56 c57 \
        c62 c63 c64 c65 c68 c69 \
        c74 c75 c76 c77 c79 c81 \
        c86 c87 c88 c89 c90 c92 \
        c98 c99 c100 c101 c103 c105 \
        c107 c109 c111 c113 c115 c116 \
        c118 c119 c121 c139 c131 c122 \
        c134 c130 c135 c140 c138 c141
do
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d0
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d1
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d2
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d3
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d4
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d5
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d6
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t0d7
  /usr/sbin/scsictl -m queue_depth=64 /dev/rdsk/${c}t1d0
done

exit
```

Appendix G Price Quotes

The following pages contain the price quotes for the hardware included in this FDR.

Juergen Mueller
HP
Cupertino, CA 95014
September 25, 2003

HP Unix Sales Development
19111 Pruneridge Avenue
Cupertino, CA 95014
(408) 447-2320

HP Integrity Superdome Enterprise Server

TPC-H Rev 2

Report Date: September 25, 2003

Description	Part Number	Source	Reference Price	Qty	Extended Price	Discount	Price
Server Hardware							
Super Dome left chassis	A5201A, Opt. 429	1	205,840	1	205,840		
Super Dome right chassis	A5202A, Opt. 429	1	218,435	1	218,435		
IPF Superdome Cell Board (sx1000)	A6866A	1	16,000	16	256,000		
3 Year Svc & Support Price (Hardware and Software)							838,197
4GB SDRAM (4x1GB DIMMS)	A6863A	1	17,750	64	1,136,000		
PCI-x I/O chassis	A6864A	1	16,805	12	201,660		
Core I/O Card	A6865A	1	1,045	1	1,045		
CPU Itanium 2, 1.5GHz w/6MB iL 3 cache (2 CPUs)	A6924A	1	40,000	32	1,280,000		
PCI 1000BT Lan Adapter	A6847A, Opt. 0D1	1	2,135	1	2,135		
PCI 2GB Fibre Channel Adapter	A6795A	1	2,240	72	161,280		
I/O chassis enclosure for PCI chassis	A5862A	1	25,725	2	51,450		
Graphite I/O expansion power subsystem	A5861D	1	34,860	1	34,860		
PCI Ultra160 SCSI Adapter	A6828A	1	1,323	1	1,323		
hp Surestore Disk System 2100	A5675A	1	700	1	700		
1-36GB LP 10K LVD SE U320 HDD	A6571A	1	720	3	2,160		
TA5300 Enclosure for DAT tape	C7508AZ	1	1,395	1	1,395		
DDS 4 tape	C5687B	1	1,450	1	1,450		
DVD Rom drive	C7499A	1	688	1	688		
SCSI Terminator LVD/SE HDTS68 Multimedia	C2364A	1	100	1	100		
HP Tape Array PSU/Fan Kit	C7496A	1	425	1	425		
SCSI Cable 10m VHDTS68/DHTS68 M/M Multimdedia	C2363B	1	335	1	335		
SCSI Cable 0.5m HDTS68 M/M Multimedia	C2978B	1	99	1	99		
SMS for HP Integrity Superdome Tower	A9801A	1	5,140	1	5,140		
				Subtotal	**3,562,520**		**838,197**
Server Software							
HPUX 11i, V2 Foundation Operating Environment	B9429AC	1	2,370	64	151,680		
				Subtotal	**151,680**		**0**
Storage							
16 meter Fibre Optic Cable	A7525A	1	260	72	18,720		
Surestore VA 7110 w/ dual controller, 1024MB mem	A7294A	1	25,440	72	1,831,680		
*(large quantity discount)							
3 Year Support Price							273,168
73GB 15K RPM FC HDD (10% sparing included)*	A7288A, Opt 0D1	1	1,702	1188	2,022,392		103,680
*(large quantity discount)							
HP Rack System/E, 41U, Quartz Color	A4902A	1	1,910	6	11,460		
HP Rear Door for 41U Quartz Rack	A5213AZ	1	334	6	2,004		
Modular Power Dist.	A5137AZ	1	145	24	3,480		
200-240 Volts Power Option	A5137AZ, Opt AW4	1	94	24	2,256		
				Subtotal	**3,891,992**		**376,848**
			Total		**7,606,192**		**1,215,045**
Large Configuration Discount and Support Prepayment*					**(3,952,665)**		**(592,022)**
			Grand Total		**3,653,527**		**623,023**

All the components in the price list are currently available. Maintenance support price is for 24 hours, 7 days with 4 hour response time